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.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (https://www.openssl.org/)

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

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/go/trademarks. 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. (1110R)

© 2019 Cisco Systems, Inc. All rights reserved.
## CONTENTS

### PREFACE
- Preface xix
  - Audience xix
  - Document Conventions xix
  - Related Documentation for Cisco Nexus 9000 Series Switches xx
  - Documentation Feedback xx
  - Communications, Services, and Additional Information xx

### CHAPTER 1
- New and Changed Information 1
  - New and Changed Information 1

### CHAPTER 2
- Overview 3
  - About Interfaces 3
    - Ethernet Interfaces 3
    - Access Ports 4
    - Routed Ports 4
    - Management Interface 4
    - Port-Channel Interfaces 4
    - Subinterfaces 4
    - Loopback Interfaces 4
    - Breakout Interfaces 4
      - Module Level Breakout 5
      - Dynamic Breakout (Per-port Level Breakout) 5
      - About the Lane Selector 6
      - Notes About Breakout Interfaces 7
    - Virtual Device Contexts 10
    - High Availability for Interfaces 10
Configuring the Throughput Delay 35
Shutting Down and Activating the Interface 36
Configuring the UDLD Mode 38
Configuring Debounce Timers 40
Configuring Port Profiles 42
  Creating a Port Profile 42
  Entering Port-Profile Configuration Mode and Modifying a Port Profile 44
  Assigning a Port Profile to a Range of Interfaces 44
  Enabling a Specific Port Profile 45
  Inheriting a Port Profile 46
  Removing a Port Profile from a Range of Interfaces 47
  Removing an Inherited Port Profile 48
Configuring 25G Autonegotiation 49
  Guidelines and Limitations for 25G Autonegotiation 49
  FEC selection with 25G Autonegotiation 49
  Enabling Autonegotiation 49
  Disabling Autonegotiation 50
Verifying the Basic Interface Parameters 51
Monitoring the Interface Counters 52
  Displaying Interface Statistics 52
  Clearing Interface Counters 53
Configuration Example for QSA 54

CHAPTER 4

Configuring Layer 2 Interfaces 55
  Information About Access and Trunk Interfaces 55
    About Access and Trunk Interfaces 55
    IEEE 802.1Q Encapsulation 57
    Access VLANs 57
    Native VLAN IDs for Trunk Ports 58
    Tagging Native VLAN Traffic 58
    Allowed VLANs 58
    Enabling 4K VLAN Configurations with switchport isolated 59
    Default Interfaces 59
    Switch Virtual Interface and Autostate Behavior 59
SVI Autostate Exclude 60
SVI Autostate Disable 60
High Availability 60
Virtualization Support 60
Counter Values 60
Licensing Requirements for Layer 2 Port Modes 61
Prerequisites for Layer 2 Interfaces 62
Guidelines and Limitations for Layer 2 Interfaces 62
Default Settings for Layer 2 Interfaces 65
Configuring Access and Trunk Interfaces 65
Guidelines for Configuring Access and Trunk Interfaces 66
Configuring a VLAN Interface as a Layer 2 Access Port 66
Configuring Access Host Ports 67
Configuring Trunk Ports 69
Configuring the Native VLAN for 802.1Q Trunking Ports 71
Configuring the Allowed VLANs for Trunking Ports 72
Configuring MAC addresses Limitation on a VLAN 74
Configuring MAC Addresses Limitation on a Port 75
Configuring switchport isolated 76
Configuring a Default Interface 77
Configuring SVI Autostate Exclude 79
Configuring SVI Autostate Disable for the System 80
Configuring SVI Autostate Disable Per SVI 81
Configuring the Device to Tag Native VLAN Traffic 83
Configuring Interface Breakout Profile for 50-G Interfaces in a 16-Slot Chassis 84
Changing the System Default Port Mode to Layer 2 85
Verifying the Interface Configuration 87
Monitoring the Layer 2 Interfaces 87
Configuration Examples for Access and Trunk Ports 88
Related Documents 88

CHAPTER 5

Configuring Layer 3 Interfaces 91
About Layer 3 Interfaces 91
Routed Interfaces 91
Contents

Subinterfaces 92
   Limitations for Subinterfaces 93
VLAN Interfaces 93
Changing VRF Membership for an Interface 93
   Notes About Changing VRF Membership for an Interface 94
Loopback Interfaces 94
IP Unnumbered 95
MAC-Embedded IPv6 Address 95
High Availability 95
Virtualization Support 95
DHCP Client 96
   Limitations for Using DHCP Client on Interfaces 96
Licensing Requirements for Layer 3 Interfaces 96
Prerequisites for Layer 3 Interfaces 97
Guidelines and Limitations for Layer 3 Interfaces 97
Default Settings 98
Configuring Layer 3 Interfaces 98
   Configuring a Routed Interface 98
   Configuring a Subinterface on a Routed Interface 100
   Configuring a Subinterface on a Port-Channel Interface 102
   Configuring a VLAN Interface 103
   Enabling Layer 3 Retention During VRF Membership Change 105
   Configuring a Loopback Interface 105
   Configuring IP Unnumbered on an Ethernet Interface 106
   Configuring OSPF for an IP Unnumbered Interface 107
   Configuring ISIS for an IP Unnumbered Interface 109
   Configuring PBR on SVI on the Gateway 110
   Configuring IP Unnumbered on SVI Secondary VLAN on the Gateway 113
   Assigning an Interface to a VRF 114
   Configuring a MAC-Embedded IPv6 Address 115
   Configuring a DHCP Client on an Interface 118
   Verifying the Layer 3 Interfaces Configuration 119
   Monitoring the Layer 3 Interfaces 120
   Configuration Examples for Layer 3 Interfaces 121
Example of Changing VRF Membership for an Interface 122
Related Documents 123

CHAPTER 6

Configuring Bidirectional Forwarding Detection 125
About BFD 125
Asynchronous Mode 125
BFD Detection of Failures 126
Distributed Operation 127
BFD Echo Function 127
Security 127
High Availability 127
Virtualization Support 128
Licensing Requirements for BFD 128
Prerequisites for BFD 128
Guidelines and Limitations 128
Default Settings 131
Configuring BFD 131
Configuration Hierarchy 131
Task Flow for Configuring BFD 132
Enabling the BFD Feature 132
Configuring Global BFD Parameters 133
Configuring BFD on an Interface 134
Configuring BFD on a Port Channel 135
Configuring the BFD Echo Function 137
Configuring Per-Member Link BFD Sessions 138
BFD Enhancement to Address Per-link Efficiency 138
Limitations of the IETF Bidirectional Forwarding Detection 139
Configuring Port Channel Interface 140
Configuring BFD Start Timer 141
Enabling IETF Per-link BFD 141
Configuring BFD Destination IP Address 142
Verifying Micro BFD Session Configurations 142
Examples: Configuring Micro BFD Sessions 143
Configuring BFD Support for Routing Protocols 146
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring BFD on BGP</td>
<td>146</td>
</tr>
<tr>
<td>Configuring BFD on EIGRP</td>
<td>147</td>
</tr>
<tr>
<td>Configuring BFD on OSPF</td>
<td>148</td>
</tr>
<tr>
<td>Configuring BFD on IS-IS</td>
<td>150</td>
</tr>
<tr>
<td>Configuring BFD on HSRP</td>
<td>151</td>
</tr>
<tr>
<td>Configuring BFD on VRRP</td>
<td>153</td>
</tr>
<tr>
<td>Configuring BFD on PIM</td>
<td>154</td>
</tr>
<tr>
<td>Configuring BFD on Static Routes</td>
<td>155</td>
</tr>
<tr>
<td>Disabling BFD on an Interface</td>
<td>156</td>
</tr>
<tr>
<td>Configuring BFD Interoperability</td>
<td>157</td>
</tr>
<tr>
<td>Configuring BFD Interoperability in Cisco NX-OS Devices in a Point-to-Point Link</td>
<td>157</td>
</tr>
<tr>
<td>Configuring BFD Interoperability in Cisco NX-OS Devices in a Switch Virtual Interface</td>
<td>158</td>
</tr>
<tr>
<td>Configuring BFD Interoperability in Cisco NX-OS Devices in Logical Mode</td>
<td>159</td>
</tr>
<tr>
<td>Verifying BFD Interoperability in a Cisco Nexus 9000 Series Device</td>
<td>160</td>
</tr>
<tr>
<td>Verifying the BFD Configuration</td>
<td>161</td>
</tr>
<tr>
<td>Monitoring BFD</td>
<td>161</td>
</tr>
<tr>
<td>BFD Multihop</td>
<td>161</td>
</tr>
<tr>
<td>BFD Multihop Number of Hops</td>
<td>162</td>
</tr>
<tr>
<td>Guidelines and Limitations for BFD Multihop</td>
<td>162</td>
</tr>
<tr>
<td>Configuring BFD Multihop Session Global Interval Parameters</td>
<td>162</td>
</tr>
<tr>
<td>Configuring Per Multihop Session BFD Parameters</td>
<td>163</td>
</tr>
<tr>
<td>Configuration Examples for BFD</td>
<td>164</td>
</tr>
<tr>
<td>Show Example for BFD</td>
<td>165</td>
</tr>
<tr>
<td>Related Documents</td>
<td>165</td>
</tr>
<tr>
<td>RFCs</td>
<td>166</td>
</tr>
</tbody>
</table>

**CHAPTER 7**

**Configuring Port Channels**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Port Channels</td>
<td>167</td>
</tr>
<tr>
<td>Port Channels</td>
<td>168</td>
</tr>
<tr>
<td>Port-Channel Interfaces</td>
<td>168</td>
</tr>
<tr>
<td>Basic Settings</td>
<td>169</td>
</tr>
<tr>
<td>Compatibility Requirements</td>
<td>170</td>
</tr>
<tr>
<td>Load Balancing Using Port Channels</td>
<td>172</td>
</tr>
<tr>
<td>Symmetric Hashing</td>
<td>173</td>
</tr>
</tbody>
</table>
Configuring the LACP Port-Channel MaxBundle 199
Configuring the LACP Fast Timer Rate 200
Configuring the LACP System Priority 202
Configuring the LACP Port Priority 203
Configuring LACP System MAC and Role 204
Disabling LACP Graceful Convergence 205
   Reenabling LACP Graceful Convergence 206
Disabling LACP Suspend Individual 207
Reenabling LACP Suspend Individual 209
Configuring Delayed LACP 210
Configuring Port Channel Hash Distribution 212
   Configuring Port Channel Hash Distribution at the Global Level 212
   Configuring Port Channel Hash Distribution at the Port Channel Level 213
Enabling ECMP Resilient Hashing 213
Disabling ECMP Resilient Hashing 214
Verifying the ECMP Resilient Hashing Configuration 215
Verifying the Port-Channel Configuration 215
Monitoring the Port-Channel Interface Configuration 216
Example Configurations for Port Channels 216
Related Documents 217

Configuring vPCs 219
Information About vPCs 219
   vPC Overview 219
Hitless vPC Role Change 221
vPC Terminology 222
   vPC Peer Link Overview 223
   Features That You Must Manually Configure on the Primary and Secondary Devices 225
   Configuring Layer 3 Backup Routes on a vPC Peer Link 226
Peer-Keepalive Link and Messages 226
vPC Peer-Gateway 227
vPC Domain 228
vPC Topology 229
Compatibility Parameters for vPC Interfaces 230
Configuration Parameters That Must Be Identical 231
Configuration Parameters That Should Be Identical 232
Consequences of Parameter Mismatches 233
vPC Number 233
Moving Other Port Channels into a vPC 233
Configuring vPC Peer Links and Links to the Core on a Single Module 234
vPC Interactions with Other Features 235
  vPC and LACP 235
  vPC Peer Links and STP 236
  vPC Peer Switch 238
  vPC and ARP or ND 238
  vPC Multicast—PIM, IGMP, and IGMP Snooping 238
Multicast PIM Dual DR (Proxy DR ) 239
  IP PIM PRE-BUILD SPT 240
  vPC Peer Links and Routing 240
Best Practices for Layer 3 and vPC Configuration 241
  CFSoE 246
  vPC and Orphan Ports 247
Virtualization Support 247
vPC Recovery After an Outage 247
  Autorecovery 247
  vPC Peer Roles After a Recovery 248
High Availability 248
  vPC Forklift Upgrade Scenario 248
Licensing Requirements for vPCs 250
Guidelines and Limitations 251
Default Settings 253
Configuring vPCs 254
  Enabling vPCs 254
  Disabling vPCs 255
Creating a vPC Domain and Entering vpc-domain Mode 256
Configuring a vPC Keepalive Link and Messages 257
Creating a vPC Peer Link 259
Configuring a vPC Peer-Gateway 261
Configuring Fast Convergence 262
Configuring LACP vPC Convergence 263
Configuring Layer 3 over vPC 264
Layer 3 over vPC Supported Topologies 266
Configuring a Graceful Consistency Check 268
Checking the Configuration Compatibility on a vPC Peer Link 269
Moving Other Port Channels into a vPC 270
Manually Configuring a vPC Domain MAC Address 272
Manually Configuring the System Priority 273
Manually Configuring the vPC Peer Device Role 274
Configuring the Tracking Feature on a Single-Module vPC 276
Configuring for Recovery After an Outage 277
Configuring Reload Restore 277
Configuring an Autorecovery 279
Configuring the Suspension of Orphan Ports 281
Configuring Delay Restore on an Orphan Port 283
Configuring the vPC Peer Switch 284
Configuring a Pure vPC Peer Switch Topology 284
Configuring a Hybrid vPC Peer Switch Topology 285
Configuring Hitless vPC Role Change 287
Use Case Scenario for vPC Role Change 288
Enabling STP to Use the Cisco MAC Address 289
Verifying the vPC Configuration 290
Monitoring vPCs 290
Configuration Examples for vPCs 290
Related Documents 293

CHAPTER 9  Configuring IP Tunnels 295
Information About IP Tunnels 295
IP Tunnel Overview 295
GRE Tunnels 296
Point-to-Point IP-in-IP Tunnel Encapsulation and Decapsulation 296
Multi-Point IP-in-IP Tunnel Decapsulation 296
Path MTU Discovery 296
CHAPTER 11

Configuring Static and Dynamic NAT Translation 337

Network Address Translation Overview 337
Information About Static NAT 338
Dynamic NAT Overview 339
Timeout Mechanisms 339
NAT Inside and Outside Addresses 340
Pool Support for Dynamic NAT 340
Static and Dynamic Twice NAT Overview 341
VRF Aware NAT 341
Guidelines and Limitations for Static NAT 343
Restrictions for Dynamic NAT 344
Guidelines and Limitations for Dynamic Twice NAT 346
Configuring Static NAT 346
   Enabling Static NAT 346
   Configuring Static NAT on an Interface 346
   Enabling Static NAT for an Inside Source Address 347
   Enabling Static NAT for an Outside Source Address 348
   Configuring Static PAT for an Inside Source Address 349
   Configuring Static PAT for an Outside Source Address 350
   Configuring Static Twice NAT 350
   Enabling and Disabling no-alias Configuration 352
   Configuration Example for Static NAT and PAT 354
   Example: Configuring Static Twice NAT 355
   Verifying the Static NAT Configuration 355
Configuring Dynamic NAT 356
   Configuring Dynamic Translation and Translation Timeouts 356
   Configuring Dynamic NAT Pool 359
   Configuring Source Lists 360
   Configuring Dynamic Twice NAT for an Inside Source Address 361
CHAPTER 12 Configuring IP Event Dampening 369
  IP Event Dampening Overview 369
  Guidelines and Limitations 369
  Interface State Change Events 370
    Suppress Threshold 370
    Half-Life Period 370
    Reuse Threshold 370
    Maximum Suppress Time 371
  Affected Components 371
    Route Types 371
    Supported Protocols 371
  How to Configure IP Event Dampening 372
    Enabling IP Event Dampening 372
    Verifying IP Event Dampening 373
    Default Settings for IP Dampening Parameters 373

CHAPTER 13 Configuring IP TCP MSS 375
  Information About IP TCP MSS 375
  Licensing Requirements for IP TCP MSS 375
  Default Settings for IP TCP MSS 376
  Guidelines and Limitations for IP TCP MSS 376
  Configuring IP TCP MSS 376
    Setting the MSS for TCP Connections 376
    Removing a Set IP TCP MSS 377
    Example: Setting the MSS for TCP Connections 377
    Example: Removing a Set IP TCP MSS 378
  Verifying IP TCP MSS 378
Preface

This preface includes the following sections:

- Audience, on page xix
- Document Conventions, on page xix
- Related Documentation for Cisco Nexus 9000 Series Switches, on page xx
- Documentation Feedback, on page xx
- Communications, Services, and Additional Information, on page xx

Audience

This publication is for network administrators who install, configure, and maintain Cisco Nexus switches.

Document Conventions

Command descriptions use the following conventions:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bold</strong></td>
<td>Bold text indicates the commands and keywords that you enter literally as shown.</td>
</tr>
<tr>
<td><em>Italic</em></td>
<td>Italic text indicates arguments for which you supply the values.</td>
</tr>
<tr>
<td>[x]</td>
<td>Square brackets enclose an optional element (keyword or argument).</td>
</tr>
<tr>
<td>[x</td>
<td>y]</td>
</tr>
<tr>
<td>{x</td>
<td>y}</td>
</tr>
<tr>
<td>[x {y</td>
<td>z}]</td>
</tr>
</tbody>
</table>
### Description Convention

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>variable</td>
<td>Indicates a variable for which you supply values, in context where italics cannot be used.</td>
</tr>
<tr>
<td>string</td>
<td>A nonquoted set of characters. Do not use quotation marks around the string or the string includes the quotation marks.</td>
</tr>
</tbody>
</table>

Examples use the following conventions:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>screen font</strong></td>
<td>Terminal sessions and information the switch displays are in screen font.</td>
</tr>
<tr>
<td><strong>boldface screen font</strong></td>
<td>Information that you must enter is in boldface screen font.</td>
</tr>
<tr>
<td><strong>italic screen font</strong></td>
<td>Arguments for which you supply values are in italic screen font.</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>Nonprinting characters, such as passwords, are in angle brackets.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Default responses to system prompts are in square brackets.</td>
</tr>
<tr>
<td>!, #</td>
<td>An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.</td>
</tr>
</tbody>
</table>

---

### Related Documentation for Cisco Nexus 9000 Series Switches

The entire Cisco Nexus 9000 Series switch documentation set is available at the following URL:


### Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus9k-docfeedback@cisco.com. We appreciate your feedback.

### Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact 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 Marketplace.
- 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.
New and Changed Information

This table summarizes the new and changed features for the Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide and where they are documented.

Table 1: New and Changed Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Changed in Release</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optics Support.</td>
<td>Added support for 10/25 LR a dual speed optical transceiver.</td>
<td>9.2(3)</td>
<td>Cisco SFP+ Adapter Module Support, on page 18</td>
</tr>
<tr>
<td>Selective Q-in-Q with Multiple Provider VLANs</td>
<td>Added support for Q-in-Q with multiple provider VLANs.</td>
<td>9.2(3)</td>
<td>Configuring Selective Q-in-Q with Multiple provider VLANs, on page 325</td>
</tr>
<tr>
<td>Breakout Support</td>
<td>Added 4x25 Gigabits breakout support for N9K-X9636C-R and N9K-X9636C-RX line cards.</td>
<td>9.2(2)</td>
<td>Notes About Breakout Interfaces, on page 7</td>
</tr>
<tr>
<td>CWDM4 Optics on 100G Interfaces</td>
<td>Added support for CWDM4 on the 36-port 100-Gigabit Ethernet QSFP28 line cards (N9K-X9636C-R), the 36-port 40-Gigabit Ethernet QSFP+ line cards (N9K-X9636Q-R), the 36-port 100-Gigabit QSFP28 line cards (N9K-X9636C-RX) and the 52-port 100-Gigabit QSFP28 line cards (N9K-X96136YC-R).</td>
<td>9.2(2)</td>
<td>Notes About Breakout Interfaces, on page 7</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Changed in Release</td>
<td>Where Documented</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>IP Event Dampening</td>
<td>Added support for IP event dampening on Cisco Nexus 9300-EX, 9300-FX, 9300-FX2, 9300-FXP, 9500-EX, and 9500-FX platform switches.</td>
<td>9.2(1)</td>
<td>Configuring IP Event Dampening, on page 369</td>
</tr>
<tr>
<td>BFD Multihop</td>
<td>Added support for BFD Multihop is supported on Cisco Nexus 9000 switches.</td>
<td>9.2(1)</td>
<td>BFD Multihop, on page 161</td>
</tr>
<tr>
<td>IP TCP MSS</td>
<td>Added support for IP TCP MSS. The IP TCP Maximum Segment Size (MSS) feature enables a switch to set a maximum segment size for all TCP connections that originate from or terminate at a Cisco Nexus 9000 Series Switches.</td>
<td>9.2(1)</td>
<td>Configuring IP TCP MSS, on page 375</td>
</tr>
<tr>
<td>TCP Aware NAT</td>
<td>Added support for TCP-aware NAT. That enables NAT flow entries to follow the state of TCP sessions and get created and deleted accordingly.</td>
<td>9.2(1)</td>
<td>Configuring FINRST and SYN Timers, on page 364</td>
</tr>
<tr>
<td>Optics Scale</td>
<td>Added one Gigabit speed support for all 48 ports on Cisco Nexus 9508 platform switches with N9K-X96136YC-R line cards.</td>
<td>9.2(1)</td>
<td>Guidelines and Limitations for Layer 2 Interfaces, on page 62</td>
</tr>
<tr>
<td>QSFP-40/100-SRBD (QSFP-100G40G-BIDI)</td>
<td>Added support for QSFP-100-G/40-G BiDi on Cisco Nexus 9500 Switches with N9K-X9636C-RX and N9K-X96136YC-R line cards to operate at a speed of 100G and interoperable with other 100G or 40G QSFPs.</td>
<td>9.2(1)</td>
<td>Guidelines and Limitations, on page 128</td>
</tr>
</tbody>
</table>
Overview

• About Interfaces, on page 3
• Virtual Device Contexts, on page 10
• High Availability for Interfaces, on page 10

About Interfaces

Cisco NX-OS supports multiple configuration parameters for each of the interface types supported. Most of these parameters are covered in this guide but some are described in other documents.

The following table shows where to get further information on the parameters you can configure for an interface.

Table 2: Interface Parameters

<table>
<thead>
<tr>
<th>Feature</th>
<th>Parameters</th>
<th>Further Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic parameters</td>
<td>description, duplex, error disable, flow control, MTU, beacon</td>
<td>“Configuring Basic Interface Parameters”</td>
</tr>
<tr>
<td>Layer 3</td>
<td>medium, IPv4 and IPv6 addresses</td>
<td>“Configuring Layer 3 Interfaces”</td>
</tr>
<tr>
<td>Layer 3</td>
<td>bandwidth, delay, IP routing, VRFs</td>
<td>Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cisco Nexus 9000 Series NX-OS Multicast Routing Configuration Guide</td>
</tr>
<tr>
<td>Port Channels</td>
<td>channel group, LACP</td>
<td>“Configuring Port Channels”</td>
</tr>
<tr>
<td>Security</td>
<td>EOU</td>
<td>Cisco Nexus 9000 Series NX-OS Security Configuration Guide</td>
</tr>
</tbody>
</table>

Ethernet Interfaces

Ethernet interfaces include routed ports.
Access Ports

An access port carries traffic for one VLAN. This type of port is a Layer 2 interface only. For more information on access ports, see the “Information About Access and Trunk Interfaces” section.

Routed Ports

A routed port is a physical port that can route IP traffic to another device. A routed port is a Layer 3 interface only. For more information on routed ports, see the “Routed Interfaces” section.

Management Interface

You can use the management Ethernet interface to connect the device to a network for remote management using a Telnet client, the Simple Network Management Protocol (SNMP), or other management agents. The management port (mgmt0) is autosensing and operates in full-duplex mode at a speed of 10/100/1000 Mb/s. For more information on the management interface, see the Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide. You will also find information on configuring the IP address and default IP routing for the management interface in this document.

Port-Channel Interfaces

A port channel is a logical interface that is an aggregation of multiple physical interfaces. You can bundle up to 32 individual links to physical ports into a port channel to improve bandwidth and redundancy. You can also use port channeling to load balance traffic across these channeled physical interfaces. For more information about port-channel interfaces, see the “Configuring Port Channels” section.

Subinterfaces

You can create virtual subinterfaces on a parent interface configured as a Layer 3 interface. A parent interface can be a physical port. Subinterfaces divide the parent interface into two or more virtual interfaces on which you can assign unique Layer 3 parameters such as IP addresses and dynamic routing protocols.

Loopback Interfaces

A virtual loopback interface is a virtual interface with a single endpoint that is always up. Any packet that is transmitted over a virtual loopback interface is immediately received by that interface. Loopback interfaces emulate a physical interface. For more information about subinterfaces, see the “Loopback Interfaces” section.

Breakout Interfaces

Cisco NX-OS supports the breakout of high bandwidth 40G interfaces at the module level or at the per-port level.
Module Level Breakout

For module level breakout, the `interface breakout` command splits the high bandwidth 40G interface of a module into four 10G interfaces. The module is reloaded and the configuration for the interface is removed when the command is executed.

The following is an example of the command:

```
switch# configure terminal
switch(config)# interface breakout module 1
Module will be reloaded. Are you sure you want to continue(yes/no)? yes
```

The `no interface breakout module module_number` command undoes the breakout configuration. It puts all interfaces of the module in 40G mode and deletes the configuration for the previous 10G interfaces.

Dynamic Breakout (Per-port Level Breakout)

For dynamic breakout (per-port level breakout), the `interface breakout` command splits high bandwidth 40G ports into four 10G breakout ports and 100G ports into four 25G breakout ports. The breakout ports are identified as `Ethernet <slot>/<front-panel-port>/<breakout-port>`. For example, per-port level breakout ports can be identified as `Ethernet 1/2/1`, `Ethernet 1/2/2`, `Ethernet 1/2/3`, and `Ethernet 1/2/4`.

When one or more 40G interfaces of the module are broken-out at the per-port level, the configuration for the interfaces are removed when the command is executed.

---

A per-port level breakout does not require the module to be reloaded.

The following is an example of configuring a breakout port:

```
switch(config)# interface breakout module 1 port 1 map 10g-4x
```

The following is an example of configuring a multiple breakout ports:

```
switch(config)# interface breakout module 1 port 1-4 map 10g-4x
```

The following is an example of configuring 40G and 10G interfaces mixed together.

```
switch(config-if)# show int eth1/49 transceiver
Ethernet1/49
transceiver is present
type is QSFP-40G-SR-BD
name is CISCO-AVAGO
part number is AFBR-79EBPZ-CS2
revision is 01

switch(config-if)# show int eth1/52 transceiver
Ethernet1/52
transceiver is present
type is QSFP-Cazadero
name is CISCO-DNI
```
part number is CAZADERO-R
revision is 03
nominal bitrate is 10000 MBit/sec per channel

switch(config-if)# show int eth1/53 transceiver
Ethernet1/53
transceiver is present
type is QSFP-Cazadero
name is CISCO-DNI
part number is CAZADERO-R
revision is 03
nominal bitrate is 10000 MBit/sec per channel

switch(config)# interface breakout module 1 port 52-53 map 10g-4x
switch(config-if)# show int br | i up
mgmt0 -- up 10.122.160.192 100 1500
Eth1/49 -- eth routed up none 40G(D) -- << Running 40G
Eth1/50 -- eth routed up none 40G(D) --
Eth1/52/1 -- eth routed up none 10G(D) -- << Broken out to 10G
Eth1/53/1 -- eth routed up none 10G(D) -- << Broken out to 10G

You can undo the breakout ports with the no interface breakout command.
The following is an example to undo the breakout ports:

switch(config)# no interface breakout module 1 port 1 map 10g-4x
switch(config)#

About the Lane Selector

The lane selector is a push button switch and 4 LEDs located on the Cisco Nexus switch (left side of front panel, labeled "LS"). The push button switch and LEDs are used to indicate the status of the ports. The lane selector is supported on Cisco Nexus Series 9000 series switches and the Cisco Nexus 3164 and 3232 switches.

By default, the LEDs indicate the link/activity status of a 1 x 40G configuration. When the ports are configured as 4 x 10G, you can access the link status of each individual 10G port with the lane selector.

By pressing the lane selector push button, the port LED shows the selected lane’s link/activity status. The 1st time the push button is pressed, the first LED displays the status of the first port. Pressing the push button a 2nd time displays the status of the second port, and so on. You can display the status of each of the four ports by pressing the push button in this manner.

For example, if port 60 is configured as 4 x 10G, pressing the lane selector push button once displays the link status of 60/1/1. Pressing the push button a second time displays the link status of 60/1/2.

When you press the push button after displaying the status of the last port, all four of the LEDs should extinguish to indicate that the lane selector has returned to display the status for the default 1 x 40G configuration.

Note

A 10G breakout port's LED blinks when the beacon feature has been configured for it.
When a port is configured to be in 10G breakout mode and no lane is selected, the 40G port's LED illuminates as green even though only one of the 10G breakout ports is up.

Notes About Breakout Interfaces

Beginning with Cisco NX-OS Release 7.0(3)F2(1), the 36-port 100-Gigabit Ethernet QSFP28 line cards (N9K-X9636C-R) and 36-port 40-Gigabit Ethernet QSFP+ line cards (N9K-X9636Q-R) provide 4x10-Gigabit support.


Caveats

- When a break-out port is configured as a part of a port-channel, you need to apply the configuration twice (after write-erase/reload), to ensure the effectiveness of the port-channel.

- When you upgrade a Cisco Nexus 9000 device to Cisco NX-OS Release 7.0(3)I7(2) or later, if a QSFP port is configured with the manual breakout command and is using a QSA, the configuration of the interface Ethernet 1/50/1 is no longer supported and will need to be removed. To restore the configuration, you must manually configure the Ethernet 1/50 on the device.

- Cisco Nexus 9000 Series switches have 40G ports. When you breakout one of the 40G ports into 4x10G ports with a QSFP breakout cable, not all of the subinterfaces can be added to a port channel. The following error message is displayed:

```
switch# channel-group 99 mode active
command failed: port not compatible [Buffer boost]
```

As a workaround, configure **no buffer-boost** on all of the subinterfaces. This enables the channel-group configuration to go through.

Using the **force** keyword on the port-channel does not enable all the interfaces to be added to the port-channel even though the error message suggests using the keyword.

- Cisco Nexus 9000 switches defaults to FC-FEC mode while the FC-FEC mode is turned off on N9K-X9636C-RX line cards, by default. To enable interoperability between two such devices, you may need to manually change the FEC settings to match with each other.

High Bandwidth Interfaces

The breakout of high bandwidth interfaces (module level or per-port level) are supported only on:

- The X9636PQ, X9432PQ, X9536PQ, and X9732C-EX line cards on a Cisco Nexus 9500 Series switch.
• The Cisco Nexus 9332PQ switch.
• The Cisco Nexus 3164Q switch.

**Cisco Nexus C92160YC Switch**

For 7.0(3)I3(1) and later, the Cisco Nexus C92160YC switch provides two different modes of operation:

- **Mode 1**: 48 * 10G/25G + 4 * 40G + 2 * 100G (Default configuration)
  - Hardware profile portmode 48x25G + 2x100G + 4x40G
  - Breakout is supported in 2 * 100G ports

- **Mode 2**: 48 * 10G/25G + 4 * 100G
  - Hardware profile portmode 48x25G + 4x100G
  - Breakout is supported on 4 * 100G ports.

Use the `show running-config | grep portmode` command to display the current operation mode.

Example:

```
switch(config-if-range)# show running-config | grep portmode
hardware profile portmode 48x25G+2x100G+4x40G
```

See the installation guide for the Cisco Nexus C92160YC switches for more information. ([Install and Upgrade Guides for Cisco Nexus 9000 Series Switches](#))

With the Cisco Nexus C92160YC switch, there are two breakout modes:

- **40G to 4x10G breakout ports**
  - Enables the breakout of 40G ports into 4 X 10G ports.
  - Use the `interface breakout module 1 port x map 10g-4x` command.

- **100G to 4x25G breakout ports**
  - Enables the breakout of 100G ports into 4 X 25G ports.
  - Use the `interface breakout module 1 port x map 25g-4x` command.

**Cisco Nexus C9272Q Switch**

For 7.0(3)I3(1) and later, the Cisco Nexus C9272Q Switch provides 72 40G ports. Ports 37 - 71 support breakout interfaces.

To configure a breakout interface, use the `interface breakout module 1 port x map 10g-4x` command.

Example:

```
switch(config)# interface breakout module 1 port 38 map 10g-4x
switch(config)# show interface ethernet 1/38 capabilities | grep -i break
```
Breakout capable: yes

**Cisco Nexus C9332PQ Switch**

For 7.0(3)I3(1) and later, the Cisco Nexus C9332PQ Switch provides 24 40G ports that support breakout mode and can be connected to four 10G NIF ports on a FEX. Ports 1 - 12 and 15 - 26 are supported. (Ports 13 and 14 ports are reserved and cannot be used for breakout mode.)

---

**Note**

All FEXs are supported.

---

**Note**

Only the Cisco Nexus 9332PQ switch provides interface breakout support for FEX fabric interfaces. (7.0(3)I3(1) and later)

---

**Cisco Nexus 9000 C93180LC-EX Switch**

For 7.0(3)I7(1) and later, Cisco Nexus 9000 C93180LC-EX switch provides three different modes of operation:

- **Mode 1**: 28 x 40G + 4 x 40G/100G (Default configuration)
  - Hardware profile portmode 4x100g + 28x40g.
  - 10x4 breakout is supported on the top ports from 1 to 27 (ports 1,3,5, 7...27). If any of the top port is broken out, the corresponding bottom port becomes non-operational. For example, if port 1 is broken out port 2 becomes non-operational.
  - 1 Gigabit and 10 Gigabit QSA is supported on ports 29, 30, 31, and 32. However, QSAs on the top and bottom front panel ports must be of same speed.
  - Ports 29, 30, 31, and 32 support 10x4, 25x4, and 50x2 breakout.

- **Mode 2**: 24 x 40G + 6 x 40G/100G
  - Hardware profile portmode 6x100g + 24x40g.
  - 10x4 breakout is supported on the top ports from 1 to 23 (ports 1,3,5, 7...23). If any of the top port is broken out the corresponding bottom port becomes non-operational.
  - Ports 25, 27, 29, 30, 31, and 32 support 10x4, 25x4, and 50x2 breakout.
  - 1 Gigabit and 10 Gigabit QSA is supported on ports 29, 30, 31, and 32. However, QSAs on the top and bottom front panel ports must be of same speed.

- **Mode 3**: 18 x 40G/100G
  - Hardware profile portmode 18x100g.
  - 10x4, 25x4, and 50x2 breakout is supported on top ports from 1 to 27 (ports 1,3,5, 7...27) and on ports 29,30,31,32.
  - 1 Gigabit and 10 Gigabit QSA is supported on all the 18 ports.
Changing Mode 3 to any other mode or vice versa requires `copy running-config startup-config` command followed by `reload` command to take effect. However, moving between Modes 1 and 2 is dynamic and requires only `copy running-config startup-config` command.

Use the `show running-config | grep portmode` command to display the current operation mode. Example:

```
switch(config-if-range)# show running-config | grep portmode
hardware profile portmode 4x100G+28x40G
```

With the Cisco Nexus C93180LC-EX switch, there are three breakout modes:

- **40G to 4x10G breakout ports**
  - Enables the breakout of 40G ports into 4 X 10G ports.
  - Use the `interface breakout module 1 port x map 10g-4x` command.

- **100G to 4x25G breakout ports**
  - Enables the breakout of 100G ports into 4 X 25G ports.
  - Use the `interface breakout module 1 port x map 25g-4x` command.

- **100G to 2x50G breakout ports**
  - Enables the breakout of 100G ports into 2 X 50G ports.
  - Use the `interface breakout module 1 port x map 50g-2x` command.

---

**Virtual Device Contexts**

Cisco NX-OS can segment operating system and hardware resources into virtual device contexts (VDCs) that emulate virtual devices. The Cisco Nexus 9000 Series switch does not support multiple VDCs. All switch resources are managed in the default VDC.

**High Availability for Interfaces**

Interfaces support stateful and stateless restarts. A stateful restart occurs on a supervisor switchover. After the switchover, Cisco NX-OS applies the runtime configuration.
CHAPTER 3

Configuring Basic Interface Parameters

- About the Basic Interface Parameters, on page 11
- Licensing Requirements, on page 19
- Guidelines and Limitations, on page 19
- Default Settings, on page 21
- Configuring the Basic Interface Parameters, on page 22
- Verifying the Basic Interface Parameters, on page 51
- Monitoring the Interface Counters, on page 52
- Configuration Example for QSA, on page 54

About the Basic Interface Parameters

Description

For the Ethernet and management interfaces, you can configure the description parameter to provide a recognizable name for the interface. Using a unique name for each interface allows you to quickly identify the interface when you are looking at a listing of multiple interfaces.

For information about setting the description parameter for port-channel interfaces, see the “Configuring a Port-Channel Description” section. For information about configuring this parameter for other interfaces, see the “Configuring the Description” section.

Beacon

The beacon mode allows you to identify a physical port by flashing its link state LED with a green light. By default, this mode is disabled. To identify the physical port for an interface, you can activate the beacon parameter for the interface.

For information about configuring the beacon parameter, see the “Configuring the Beacon Mode” section.

Error Disabled

A port is in the error-disabled (err-disabled) state when the port is enabled administratively (using the no shutdown command) but disabled at runtime by any process. For example, if UDLD detects a unidirectional link, the port is shut down at runtime. However, because the port is administratively enabled, the port status
displays as err-disable. Once a port goes into the err-disable state, you must manually reenable it or you can configure a timeout value that provides an automatic recovery. By default, the automatic recovery is not configured, and by default, the err-disable detection is enabled for all causes.

When an interface is in the err-disabled state, use the `errdisable detect cause` command to find information about the error.

You can configure the automatic error-disabled recovery timeout for a particular error-disabled cause and configure the recovery period.

The `errdisable recovery cause` command provides an automatic recovery after 300 seconds.

You can use the `errdisable recovery interval` command to change the recovery period within a range of 30 to 65535 seconds. You can also configure the recovery timeout for a particular err-disable cause.

If you do not enable the error-disabled recovery for the cause, the interface stays in the error-disabled state until you enter the `shutdown` and `no shutdown` commands. If the recovery is enabled for a cause, the interface is brought out of the error-disabled state and allowed to retry operation once all the causes have timed out. Use the `show interface status err-disabled` command to display the reason behind the error.

### Interface Status Error Policy

Cisco NX-OS policy servers such as Access Control List (ACL) Manager and Quality of Service (QoS) Manager, maintain a policy database. A policy is defined through the command-line interface.

Policies are pushed when you configure a policy on an interface to ensure that policies that are pushed are consistent with the hardware policies. To clear the errors and to allow the policy programming to proceed with the running configuration, enter the `no shutdown` command. If the policy programming succeeds, the port is allowed to come up. If the policy programming fails, the configuration is inconsistent with the hardware policies and the port is placed in an error-disabled policy state. The error-disabled policy state remains and the information is stored to prevent the same port from being brought up in the future. This process helps to avoid unnecessary disruption to the system.

### Port MTU Size

The maximum transmission unit (MTU) size specifies the maximum frame size that an Ethernet port can process. For transmissions to occur between two ports, you must configure the same MTU size for both ports. A port drops any frames that exceed its MTU size.

By default, each port has an MTU of 1500 bytes, which is the IEEE 802.3 standard for Ethernet frames. Larger MTU sizes are possible for more efficient processing of data with less overhead. The larger frames, called jumbo frames, can be up to 9216 bytes in size, which is also the default system jumbo MTU size.

On a Layer 3 interface, you can configure an MTU size between 576 and 9216 bytes.

**Note**

The global LAN port MTU size applies to the traffic through a Layer 3 Ethernet LAN port that is configured with a nondefault MTU size.

For a Layer 2 port, you can configure an MTU size that is either the system default (1500 bytes) or the system jumbo MTU size (initially 9216 bytes).
If you change the system jumbo MTU size, Layer 2 ports automatically use the system default MTU size (1500 bytes) unless you specify the new system jumbo MTU size for some or all of those ports.

For the Cisco Nexus 9372 switch, the following applies:

- The 10-G interfaces are mapped to specific hardware ports where the default MTU is 1500.
- The 40-G interfaces are mapped as a HiGiG port where the default MTU is 3FFF and the MTU limit check is disabled.
- In the case of 40-G interfaces, since the MTU limit check is disabled, it ignores the packet size and traffic flows irrespective of its MTU.

For information about setting the MTU size, see the “Configuring the MTU Size” section.

**Bandwidth**

Ethernet ports have a fixed bandwidth of 1,000,000 Kb at the physical layer. Layer 3 protocols use a bandwidth value that you can set for calculating their internal metrics. The value that you set is used for informational purposes only by the Layer 3 protocols—it does not change the fixed bandwidth at the physical layer. For example, the Enhanced Interior Gateway Routing Protocol (EIGRP) uses the minimum path bandwidth to determine a routing metric, but the bandwidth at the physical layer remains at 1,000,000 Kb.

For information about configuring the bandwidth parameter for port-channel interfaces, see the “Configuring the Bandwidth and Delay for Informational Purposes” section. For information about configuring the bandwidth parameter for other interfaces, see the “Configuring the Bandwidth” section.

**Throughput Delay**

Specifying a value for the throughput-delay parameter provides a value used by Layer 3 protocols; it does not change the actual throughput delay of an interface. The Layer 3 protocols can use this value to make operating decisions. For example, the Enhanced Interior Gateway Routing Protocol (EIGRP) can use the delay setting to set a preference for one Ethernet link over another, if other parameters such as link speed are equal. The delay value that you set is in the tens of microseconds.

For information about configuring the bandwidth parameter for port-channel interfaces, see the “Configuring the Bandwidth and Delay for Informational Purposes” section. For information about configuring the throughput-delay parameter for other interfaces, see the “Configuring the Throughput Delay” section.

**Administrative Status**

The administrative-status parameter determines whether an interface is up or down. When an interface is administratively down, it is disabled and unable to transmit data. When an interface is administratively up, it is enabled and able to transmit data.

For information about configuring the administrative status parameter for port-channel interfaces, see the “Shutting Down and Restarting the Port-Channel Interface” section. For information about configuring the administrative-status parameter for other interfaces, see the “Shutting Down and Activating the Interface” section.
Unidirectional Link Detection Parameter

UDLD Overview

The Cisco-proprietary Unidirectional Link Detection (UDLD) protocol allows devices that are connected through fiber-optic or copper (for example, Category 5 cabling) Ethernet cables to monitor the physical configuration of the cables and detect when a unidirectional link exists. When a device detects a unidirectional link, UDLD shuts down the affected LAN port and alerts the user. Unidirectional links can cause a variety of problems.

UDLD performs tasks that autonegotiation cannot perform, such as detecting the identities of neighbors and shutting down misconnected LAN ports. When you enable both autonegotiation and UDLD, Layer 1 detections work to prevent physical and logical unidirectional connections and the malfunctioning of other protocols.

A unidirectional link occurs whenever traffic transmitted by the local device over a link is received by the neighbor but traffic transmitted from the neighbor is not received by the local device. If one of the fiber strands in a pair is disconnected, as long as autonegotiation is active, the link does not stay up. In this case, the logical link is undetermined, and UDLD does not take any action. If both fibers are working normally at Layer 1, UDLD determines whether those fibers are connected correctly and whether traffic is flowing bidirectionally between the correct neighbors. This check cannot be performed by autonegotiation, because autonegotiation operates at Layer 1.

The Cisco Nexus 9000 Series device periodically transmits UDLD frames to neighbor devices on LAN ports with UDLD enabled. If the frames are echoed back within a specific time frame and they lack a specific acknowledgment (echo), the link is flagged as unidirectional and the LAN port is shut down. Devices on both ends of the link must support UDLD in order for the protocol to successfully identify and disable unidirectional links. You can configure the transmission interval for the UDLD frames, either globally or for the specified interfaces.

By default, UDLD is locally disabled on copper LAN ports to avoid sending unnecessary control traffic on this type of media.

The figure shows an example of a unidirectional link condition. Device B successfully receives traffic from device A on the port. However, device A does not receive traffic from device B on the same port. UDLD detects the problem and disables the port.

Figure 1: Unidirectional Link

![Diagram of Unidirectional Link]
Default UDLD Configuration

The following table shows the default UDLD configuration.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDLD global enable state</td>
<td>Globally disabled</td>
</tr>
<tr>
<td>UDLD per-port enable state for fiber-optic media</td>
<td>Enabled on all Ethernet fiber-optic LAN ports</td>
</tr>
<tr>
<td>UDLD per-port enable state for twisted-pair (copper) media</td>
<td>Disabled on all Ethernet 10/100 and 1000BASE-TX LAN ports</td>
</tr>
<tr>
<td>UDLD aggressive mode</td>
<td>Disabled</td>
</tr>
<tr>
<td>UDLD message interval</td>
<td>15 seconds</td>
</tr>
</tbody>
</table>

For information about configuring the UDLD for the device and its port, see the “Configuring the UDLD Mode” section.

**UDLD Aggressive and Nonaggressive Modes**

UDLD aggressive mode is disabled by default. You can configure UDLD aggressive mode only on point-to-point links between network devices that support UDLD aggressive mode. If UDLD aggressive mode is enabled, when a port on a bidirectional link that has a UDLD neighbor relationship established stops receiving UDLD frame, UDLD tries to reestablish the connection with the neighbor. After eight failed retries, the port is disabled.

When you enable the UDLD aggressive mode, the following occurs:

- One side of a link has a port stuck (both transmission and receive)
- One side of a link remains up while the other side of the link is down

In these cases, the UDLD aggressive mode disables one of the ports on the link, which prevents traffic from being discarded.

---

**Note**

You enable the UDLD aggressive mode globally to enable that mode on all the fiber ports. You must enable the UDLD aggressive mode on copper ports on specified interfaces.

---

**Tip**

When a line card upgrade is being performed during an in-service software upgrade (ISSU) and some of the ports on the line card are members of a Layer 2 port channel and are configured with UDLD aggressive mode, if you shut down one of the remote ports, UDLD puts the corresponding port on the local device into an error-disabled state. This behavior is correct.

To restore service after the ISSU has completed, enter the `shutdown` command followed by the `no shutdown` command on the local port.
Port-Channel Parameters

A port channel is an aggregation of physical interfaces that comprise a logical interface. You can bundle up to 32 individual interfaces into a port channel to provide increased bandwidth and redundancy. Port channeling also load balances traffic across these physical interfaces. The port channel stays operational if at least one physical interface within the port channel is operational.

You can create Layer 3 port channels by bundling compatible Layer 3 interfaces.

Any configuration changes that you apply to the port channel are applied to each interface member of that port channel.

For information about port channels and for information about configuring port channels, see Chapter 6, “Configuring Port Channels.”

Port Profiles

On Cisco Nexus 9300 Series switches, you can create a port profile that contains many interface commands and apply that port profile to a range of interfaces. Each port profile can be applied only to a specific type of interface; the choices are as follows:

- Ethernet
- VLAN network interface
- Port channel

When you choose Ethernet or port channel as the interface type, the port profile is in the default mode which is Layer 3. Enter the switchport command to change the port profile to Layer 2 mode.

You inherit the port profile when you attach the port profile to an interface or range of interfaces. When you attach, or inherit, a port profile to an interface or range of interfaces, the system applies all the commands in that port profile to the interfaces. Additionally, you can have one port profile inherit the settings from another port profile. Inheriting another port profile allows the initial port profile to assume all of the commands of the second, inherited, port profile that do not conflict with the initial port profile. Four levels of inheritance are supported. The same port profile can be inherited by any number of port profiles.

The system applies the commands inherited by the interface or range of interfaces according to the following guidelines:

- Commands that you enter under the interface mode take precedence over the port profile’s commands if there is a conflict. However, the port profile retains that command in the port profile.

- The port profile’s commands take precedence over the default commands on the interface, unless the port-profile command is explicitly overridden by the default command.

- When a range of interfaces inherits a second port profile, the commands of the initial port profile override the commands of the second port profile if there is a conflict.

- After you inherit a port profile onto an interface or range of interfaces, you can override individual configuration values by entering the new value at the interface configuration level. If you remove the individual configuration values at the interface configuration level, the interface uses the values in the port profile again.

- There are no default configurations associated with a port profile.
A subset of commands are available under the port-profile configuration mode, depending on which interface type you specify.

---

**Note**

You cannot use port profiles with Session Manager. See the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide* for information about Session Manager.

---

To apply the port-profile configurations to the interfaces, you must enable the specific port profile. You can configure and inherit a port profile onto a range of interfaces prior to enabling the port profile. You would then enable that port profile for the configurations to take effect on the specified interfaces.

If you inherit one or more port profiles onto an original port profile, only the last inherited port profile must be enabled; the system assumes that the underlying port profiles are enabled.

When you remove a port profile from a range of interfaces, the system undoes the configuration from the interfaces first and then removes the port-profile link itself. Also, when you remove a port profile, the system checks the interface configuration and either skips the port-profile commands that have been overridden by directly entered interface commands or returns the command to the default value.

If you want to delete a port profile that has been inherited by other port profiles, you must remove the inheritance before you can delete the port profile.

You can also choose a subset of interfaces from which to remove a port profile from among that group of interfaces that you originally applied the profile. For example, if you configured a port profile and configured ten interfaces to inherit that port profile, you can remove the port profile from just some of the specified ten interfaces. The port profile continues to operate on the remaining interfaces to which it is applied.

If you delete a specific configuration for a specified range of interfaces using the interface configuration mode, that configuration is also deleted from the port profile for that range of interfaces only. For example, if you have a channel group inside a port profile and you are in the interface configuration mode and you delete that port channel, the specified port channel is also deleted from the port profile as well.

Just as in the device, you can enter a configuration for an object in port profiles without that object being applied to interfaces yet. For example, you can configure a virtual routing and forwarding (VRF) instance without it being applied to the system. If you then delete that VRF and related configurations from the port profile, the system is unaffected.

After you inherit a port profile on an interface or range of interfaces and you delete a specific configuration value, that port-profile configuration is not operative on the specified interfaces.

If you attempt to apply a port profile to the wrong type of interface, the system returns an error.

When you attempt to enable, inherit, or modify a port profile, the system creates a checkpoint. If the port-profile configuration fails, the system rolls back to the prior configuration and returns an error. A port profile is never only partially applied.

---

**Cisco QSFP+ to SFP+ Adapter Module Support**

The Cisco QSFP+ to SFP+ Adapter (QSA) module provides 10G support for the 40G uplink ports that are a part of the Cisco Nexus M6PQ and Cisco Nexus M12PQ uplink modules of specific Cisco Nexus 9300 devices.

A group of six consecutive ports in the M6PQ or M12PQ uplink module must be operating at the same speed (40G or 10G) to use the QSA/QSFP modules.
• For Cisco Nexus 9396PX devices, 2/1-6 ports form the first port speed group and the remaining 2/7-12 ports form the second port speed group.

• For Cisco Nexus 93128PX/TX devices, 2/1-6 ports form the first port speed group and the remaining 2/7-8 ports form the second port speed group.

• For Cisco Nexus 937xPX/TX devices, 1/49-54 ports form the only port speed group.

• For Cisco Nexus 93120TX devices, 1/97-102 ports form the only port speed group.

• For Cisco Nexus 9332PQ devices, 1/27-32 ports form the only port speed group.

Use the `speed-group 10000` command to configure the first port of a port speed group for the QSA. This command specifies the administrator speed preference for the port group. (The default port speed is 40G.)

• The `speed-group 10000` command specifies a speed of 10G.

• The `no speed-group 10000` command specifies a speed of 40G.

• Uplink modules should not be removed from a Cisco Nexus 9300 platform switch that is running Cisco NX-OS Release 7.0(3)I7(5). The ports on uplink modules should be used only for uplinks.

• Beginning with Cisco NX-OS Release 9.2(2), CWDM4 is supported on the 36-port 100-Gigabit Ethernet QSFP28 line cards (N9K-X9636C-R), the 36-port 40-Gigabit Ethernet QSFP+ line cards (N9K-X9636Q-R), the 36-port 100-Gigabit QSFP28 line cards (N9K-X9636C-RX) and the 52-port 100-Gigabit QSFP28 line cards (N9K-X96136YC-R).

After the speed has been configured, the compatible transceiver modules are enabled. The remaining transceiver modules in the port group (incompatible transceiver modules) become error disabled with a reason of "check speed-group config".

---

**Note**
The Cisco QSFP+ to SFP+ Adapter (QSA) module does not provide 10G support for the 40G line cards for Cisco Nexus 9500 devices.

You can use a QSFP-to-SFP adapter on Cisco Nexus 9200 and 9300-EX Series switches and Cisco Nexus 3232C and 3264Q Series switches.

---

**Cisco SFP+ Adapter Module Support**

You can use the CVR-2QSFP28-8SFP adapter for 25-Gigabit optics support on 100-Gigabit ports of the Cisco Nexus 9236C switch.

The `interface breakout module` command can be used to split this switch's 100G interfaces into four 25G interfaces. After you enter this command, you must copy the running configuration to the startup configuration.

Beginning Cisco NX-OS Release 9.2(3), 10/25 LR is supported on N9K-C93180YC-EX, N9K-X97160YC-EX, N9K-C93180YC-FX, N9K-C93240YC-FX2 and N3K-C34180YC switches. This dual speed optical transceiver operates at 25G by default and it seamlessly inter-operates with other 25G LR transceivers. Because auto speed sensing is not supported on this device, to interoperate with a 10G transceiver, you must manually configure it to use 10G speed.
Licensing Requirements

The following table shows the licensing requirements for this feature:

<table>
<thead>
<tr>
<th>Product</th>
<th>License Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco NX-OS</td>
<td>The basic interface parameters require no license. Any feature not included in a license package is bundled with the NX-OS image and is provided at no extra charge to you.</td>
</tr>
</tbody>
</table>

Guidelines and Limitations

Basic interface parameters have the following configuration guidelines and limitations:

- MDIX is enabled by default on copper ports. It is not possible to disable it.

- **show** commands with the `internal` keyword are not supported.

- Fiber-optic Ethernet ports must use Cisco-supported transceivers. To verify that the ports are using Cisco-supported transceivers, use the `show interface transceivers` command. Interfaces with Cisco-supported transceivers are listed as functional interfaces.

- A port can be either a Layer 2 or a Layer 3 interface; it cannot be both simultaneously.
  
  By default, each port is a Layer 3 interface.

  You can change a Layer 3 interface into a Layer 2 interface by using the `switchport` command. You can change a Layer 2 interface into a Layer 3 interface by using the `no switchport` command.

- You usually configure Ethernet port speed and duplex mode parameters to auto to allow the system to negotiate the speed and duplex mode between ports. If you decide to configure the port speed and duplex modes manually for these ports, consider the following:
  
  - Before you configure the speed and duplex mode for an Ethernet or management interface, see the Default Settings section for the combinations of speeds and duplex modes that can be configured at the same time.
  
  - If you set the Ethernet port speed to auto, the device automatically sets the duplex mode to auto.
  
  - If you enter the **no speed** command, the device automatically sets both the speed and duplex parameters to auto (the **no speed** command produces the same results as the **speed auto** command).
  
  - If you configure an Ethernet port speed to a value other than auto (for example, 1G, 10G, or 40G), you must configure the connecting port to match. Do not configure the connecting port to negotiate the speed.
  
  - To configure speed, duplex, and automatic flow control for an Ethernet interface, you can use the `negotiate auto` command. To disable automatic negotiation, use the `no negotiate auto` command.
The device cannot automatically negotiate the Ethernet port speed and duplex mode if the connecting port is configured to a value other than auto.

Changing the Ethernet port speed and duplex mode configuration might shut down and reenable the interface.

- When using a QSFP-40G-CR4 cable to connect between N9K-C9332PQ non-ALE ports and N9K-C9372PX ALE ports, you need to manually set the speed to 40000.
- For BASE-T copper ports, auto-negotiation is enabled even when fixed speed is configured.
- Using a regular expression to address a set of interfaces is supported with the `regex` command option. The `regex` command option is an extension that is available for all interface commands.

Example:

```
switch(config-if-range)# interface ethernet regex [2]/
switch(config-if-range)# where
   conf; interface Ethernet2/1-8   admin@switch%default
switch(config-if-range)# interface ethernet regex [1]/2[2-4]
switch(config-if-range)# where
   conf; interface Ethernet1/22-24 admin@switch%default
```

- The source-interface command option provides support for management applications to configure an IPv4 and/or IPv6 inband or outband source IP address for the copy command and other processes (such as tacacs, ntp, ping/ping6, icmp-error and traceroute).

  - Configuration commands
    ```
    ip services source-interface interface vrf vrf name
    ```
    Examples:
    ```
    • ip ftp source-interface ethernet 8/1 vrf management
    • ip http source-interface loopback 1 vrf blue
    • ip ssh source-interface ethernet ethernet 5/1
      /*This command executes in the VRF context.*/
    • ip ping source-interface ethernet 8/1 vrf blue
    • ip traceroute source-interface ethernet 8/1 vrf red
    • ip icmp-errors source-interface ethernet 8/1
      /*This command executes in the VRF context.*/
    ```

  - Show commands:
    ```
    show ip copy services source-interface interface vrf vrf name
    ```
    ```
    • show ip ftp source-interface ethernet 8/1 vrf management
    • show ip http source-interface loopback 1 vrf blue
    ```
• show ip ssh source-interface ethernet ethernet 5/1
  /*This command executes in the VRF context.*/
• show ip ping source-interface ethernet 8/1 vrf blue
• show ip traceroute source-interface ethernet 8/1 vrf red
• show ip icmp-errors source-interface ethernet 8/1
  /*This command executes in the VRF context.*/

• Service commands:

  copy service://username@hostname/path file source-interface interface name

  Examples:
  • copy ftp://username@hostname/usr/local/bin file source-interface ethernet 8/1
  • copy scp://username@hostname/usr/local/bin file source-interface ethernet 8/1
  • copy tftp://username@hostname/usr/local/bin file source-interface ethernet 8/1
  • copy http://username@hostname/usr/local/bin file source-interface ethernet 8/1
  • copy sftp://username@hostname/usr/local/bin file source-interface ethernet 8/1

• Port profiles are supported on Cisco Nexus 9300 Series switches and Cisco Nexus 9500 Series switches.
• Auto-negotiation is not supported on 25-Gigabit Ethernet Transceiver Modules on Cisco Nexus 9200 and 9300-EX platform switches; and Cisco Nexus 9500 platform switches that uses N9K-X9700-EX line cards.
• Autonegotiation is not supported on Cisco Nexus N9K-C92300YC switch.
• Autonegotiation is not supported on 25G breakout ports.

## Default Settings

The following lists the default settings for the basic interface parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Blank</td>
</tr>
<tr>
<td>Beacon</td>
<td>Disabled</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>Data rate of interface</td>
</tr>
<tr>
<td>Throughput delay</td>
<td>100 microseconds</td>
</tr>
</tbody>
</table>
Configuring the Basic Interface Parameters

When you configure an interface, you must specify the interface before you can configure its parameters.

Specifying the Interfaces to Configure

Before you begin

Before you can configure the parameters for one or more interfaces of the same type, you must specify the type and the identities of the interfaces.

The following table shows the interface types and identities that you should use for specifying the Ethernet and management interfaces.

Table 4: Information Needed to Identify an Interface for Configurations

<table>
<thead>
<tr>
<th>Interface Type</th>
<th>Identity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet</td>
<td>I/O module slot numbers and port numbers on the module</td>
</tr>
<tr>
<td>Management</td>
<td>0 (for port 0)</td>
</tr>
</tbody>
</table>
The interface range configuration mode allows you to configure multiple interfaces with the same configuration parameters. After you enter the interface range configuration mode, all command parameters you enter are attributed to all interfaces within that range until you exit out of the interface range configuration mode.

You enter a range of interfaces using dashes (-) and commas (,). Dashes separate contiguous interfaces and commas separate noncontiguous interfaces. When you enter noncontiguous interfaces, you must enter the media type for each interface.

This example shows how to configure a contiguous interface range:

```
switch(config)# interface ethernet 2/29-30
switch(config-if-range)#
```

This example shows how to configure a noncontiguous interface range:

```
switch(config)# interface ethernet 2/29, ethernet 2/33, ethernet 2/35
switch(config-if-range)#
```

You can specify subinterfaces in a range only when the subinterfaces are on the same port, for example, 2/29.1-2. But you cannot specify the subinterfaces in a range of ports, for example, you cannot enter 2/29.2-30.2. You can specify two of the subinterfaces discretely, for example, you can enter 2/29.2, 2/30.2.

This example shows how to configure a breakout cable:

```
switch(config)# interface ethernet 1/2/1
switch(config-if-range)#
```

<table>
<thead>
<tr>
<th>SUMMARY STEPS</th>
<th>DETAILED STEPS</th>
</tr>
</thead>
</table>
| **1. configure terminal** | **Step 1** configure terminal  
Example:  
switch# configure terminal  
switch(config)# |

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
| Example:  
switch# configure terminal  
switch(config)# | |

| Step 2 | interface interface  
Example:  
switch(config)# interface ethernet 2/1  
switch(config-if)#  
Example:  
switch(config)# interface mgmt0  
switch(config-if)# | Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use ethernet slot/port. For the management interface, use mgmt0.  
Examples:  
• The 1st example shows how to specify the slot 2, port 1 Ethernet interface.  
• The 2nd example shows how to specify the management interface. |

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Command or Action</th>
</tr>
</thead>
</table>
| **1. configure terminal** | **Step 1** configure terminal  
Example:  
switch# configure terminal  
switch(config)# |

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
| Example:  
switch# configure terminal  
switch(config)# | |

| Step 2 | interface interface  
Example:  
switch(config)# interface ethernet 2/1  
switch(config-if)#  
Example:  
switch(config)# interface mgmt0  
switch(config-if)# | Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use ethernet slot/port. For the management interface, use mgmt0.  
Examples:  
• The 1st example shows how to specify the slot 2, port 1 Ethernet interface.  
• The 2nd example shows how to specify the management interface. |

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Command or Action</th>
</tr>
</thead>
</table>
| **1. configure terminal** | **Step 1** configure terminal  
Example:  
switch# configure terminal  
switch(config)# |

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
| Example:  
switch# configure terminal  
switch(config)# | |
### Configuring the Description

You can provide textual interface descriptions for the Ethernet and management interfaces.

**SUMMARY STEPS**

1. configure terminal
2. interface interface
3. description text
4. show interface interface
5. exit
6. copy running-config startup-config

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> switch(config)# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface interface</td>
<td>Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use <code>ethernet slot/port</code>. For the management interface, use <code>mgmt0</code>.</td>
</tr>
<tr>
<td><strong>Example:</strong> switch(config)# interface ethernet 2/1</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> switch(config)# interface mgmt0</td>
<td>Examples:</td>
</tr>
<tr>
<td><strong>Step 3</strong> description text</td>
<td>Specifies the description for the interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
</tbody>
</table>

---

You do not need to add a space between the interface type and identity (port or slot/port number). For example, for the Ethernet slot 4, port 5 interface, you can specify either “ethernet 4/5” or “ethernet4/5.” The management interface is either “mgmt0” or “mgmt 0.”

When you are in the interface configuration mode, the commands that you enter configure the interface that you specified for this mode.
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>switch(config-if)# description Ethernet port 3 on module 1</code></td>
<td>(Optional) Displays the interface status, which includes the description parameter.</td>
</tr>
<tr>
<td><code>Step 4 show interface interface</code></td>
<td>Exits the interface mode.</td>
</tr>
<tr>
<td>Example: <code>switch(config-if)# show interface ethernet 2/1</code></td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td><code>Step 5 exit</code></td>
<td>Exits the interface mode.</td>
</tr>
<tr>
<td>Example: <code>switch(config-if)# exit switch(config)#</code></td>
<td></td>
</tr>
<tr>
<td><code>Step 6 copy running-config startup-config</code></td>
<td>Exits the interface mode.</td>
</tr>
<tr>
<td>Example: <code>switch(config)# copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>

### Example

This example shows how to set the interface description to Ethernet port 24 on module 3:

```
switch# configure terminal
switch(config)# interface ethernet 3/24
switch(config-if)# description server1
switch(config-if)#
```

The output of the `show interface eth` command is enhanced as shown in the following example:

```
Switch# show version
Software
BIOS: version 06.26
NXOS: version 6.1(2)I2(1) [build 6.1(2)I2.1]
BIOS compile time: 01/15/2014
NXOS image file is: bootflash://n9000-dk9.6.1.2.I2.1.bin
NXOS compile time: 2/25/2014 2:00:00 [02/25/2014 10:39:03]

switch# show interface ethernet 6/36
Ethernet6/36 is up
admin state is up, Dedicated Interface
Hardware: 40000 Ethernet, address: 0022.bdf6.bf91 (bia 0022.bdf8.2bf3)
Internet Address is 192.168.100.1/24
MTU 9216 bytes, BW 4000000 Kbit, DLY 10 usec
```

### Configuring the Beacon Mode

You can enable the beacon mode for an Ethernet port to flash its LED to confirm its physical location.

**SUMMARY STEPS**

1. `configure terminal`
### Configuring the Beacon Mode

2. `interface ethernet slot/port`
3. `[no] beacon`
4. `show interface ethernet slot/port`
5. `exit`
6. `copy running-config startup-config`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td>Example:</td>
</tr>
<tr>
<td><code>switch# configure terminal</code></td>
<td><code>switch(config)#</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Specifies an interface to configure, and enters interface configuration mode.</td>
</tr>
<tr>
<td><code>interface ethernet slot/port</code></td>
<td>Example:</td>
</tr>
<tr>
<td><code>switch(config)# interface ethernet 3/1</code></td>
<td><code>switch(config-if)#</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enables the beacon mode or disables the beacon mode. The default mode is disabled.</td>
</tr>
<tr>
<td><code>[no] beacon</code></td>
<td>Example:</td>
</tr>
<tr>
<td><code>switch(config)# beacon</code></td>
<td><code>switch(config-if)#</code></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>(Optional) Displays the interface status, which includes the beacon mode state.</td>
</tr>
<tr>
<td><code>show interface ethernet slot/port</code></td>
<td>Example:</td>
</tr>
<tr>
<td><code>switch(config)# show interface ethernet 2/1</code></td>
<td><code>switch(config-if)#</code></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Exits the interface mode.</td>
</tr>
<tr>
<td><code>exit</code></td>
<td>Example:</td>
</tr>
<tr>
<td><code>switch(config-if)# exit</code></td>
<td><code>switch(config)#</code></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td><code>copy running-config startup-config</code></td>
<td>Example:</td>
</tr>
<tr>
<td><code>switch(config)# copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>

### Example

This example shows how to enable the beacon mode for the Ethernet port 3/1:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# beacon
switch(config-if)#
```

This example shows how to disable the beacon mode for the Ethernet port 3/1:

```
switch(config)# interface ethernet 3/1
switch(config-if)# no beacon
switch(config-if)#
```
This example shows how to configure the dedicated mode for Ethernet port 4/17 in the group that includes ports 4/17, 4/19, 4/21, and 4/23:

```
switch# configure terminal
switch(config)# interface ethernet 4/17, ethernet 4/19, ethernet 4/21, ethernet 4/23
switch(config-if)# shutdown
switch(config-if)# interface ethernet 4/17
switch(config-if)# no shutdown
```

### Configuring the Error-Disabled State

You can view the reason that an interface moves to the error-disabled state and configure automatic recovery.

**Enabling the Error-Disable Detection**

You can enable error-disable detection in an application. As a result, when a cause is detected on an interface, the interface is placed in an error-disabled state, which is an operational state that is similar to the link-down state.

**SUMMARY STEPS**

1. configure terminal
2. errdisable detect cause {acl-exception | all | link-flap | loopback}
3. shutdown
4. no shutdown
5. show interface status err-disabled
6. copy running-config startup-config

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure terminal</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 2**        | Specifies a condition under which to place the interface in an error-disabled state. The default is enabled. |
| errdisable detect cause {acl-exception | all | link-flap | loopback} | |
| Example:          |         |
| switch(config)# errdisable detect cause all |
| switch(config-if)# |         |
Enabling the Error-Disabled Recovery

You can specify the application to bring the interface out of the error-disabled state and retry coming up. It retries after 300 seconds, unless you configure the recovery timer (see the `errdisable recovery interval` command).

### SUMMARY STEPS

1. configure terminal
2. errdisable recovery cause {all | bpduguard | failed-port-state | link-flap | loopback | miscabling | psecure-violation | security-violation | storm-control | udl | vpc-peerlink}
3. show interface status err-disabled
4. copy running-config startup-config

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
</tr>
</thead>
</table>
| switch# configure terminal  
  switch(config)# |

### Step 2

#### errdisable recovery cause

Specifies a condition under which the interface automatically recovers from the error-disabled state, and the device retries bringing the interface up. The device waits 300 seconds to retry. The default is disabled.

<table>
<thead>
<tr>
<th>Command or Action</th>
</tr>
</thead>
</table>
| switch(config)# errdisable recovery cause all  
  switch(config-if)# |

#### Example:

This example shows how to enable error-disabled recovery under all conditions:

```
switch(config)# errdisable recovery cause all  
switch(config)#
```

### Configuring the Error-Disabled Recovery Interval

You can configure the error-disabled recovery timer value.

### SUMMARY STEPS

1. configure terminal
2. errdisable recovery interval interval
3. show interface status err-disabled
4. copy running-config startup-config

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
</tr>
</thead>
</table>
| Step 1  
  configure terminal  
  Example:  
  switch# configure terminal  
  switch(config)# |

Purpose

<table>
<thead>
<tr>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

---

Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide, Release 9.2(x)
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong></td>
<td>errdisable recovery interval interval</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>specifies the interval for the interface to recover from the error-disabled state. The range is from 30 to 65535 seconds, and the default is 300 seconds.</td>
</tr>
<tr>
<td>switch(config)# errdisable recovery interval 32</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>show interface status err-disabled</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>(Optional) Displays information about error-disabled interfaces.</td>
</tr>
<tr>
<td>switch(config)# show interface status err-disabled</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>copy running-config startup-config</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td>switch(config)# copy running-config startup-config</td>
<td></td>
</tr>
</tbody>
</table>

**Example**

This example shows how to configure the error-disabled recovery timer to set the interval for recovery to 32 seconds:

```
switch(config)# errdisable recovery interval 32
switch(config)#
```

**Configuring the MTU Size**

You can configure the maximum transmission unit (MTU) size for Layer 2 and Layer 3 Ethernet interfaces. For Layer 3 interfaces, you can configure the MTU to be between 576 and 9216 bytes (even values are required). For Layer 2 interfaces, you can configure the MTU to be either the system default MTU (1500 bytes) or the system jumbo MTU size (which has the default size of 9216 bytes).

**Note**

You can change the system jumbo MTU size, but if you change that value, the Layer 2 interfaces that use that value automatically changes to the new system jumbo MTU value.

By default, Cisco NX-OS configures Layer 3 parameters. If you want to configure Layer 2 parameters, you need to switch the port mode to Layer 2.

You can change the port mode by using the `switchport` command.

After changing the port mode to Layer 2, you can return to configuring Layer 3 interfaces by changing the port mode again, by using the `no switchport` command.

**Configuring the Interface MTU Size**

For Layer 3 interfaces, you can configure an MTU size that is between 576 and 9216 bytes.

For Layer 2 interfaces, you can configure all Layer 2 interfaces to use either the default MTU size (1500 bytes) or the system jumbo MTU size (default size of 9216 bytes).
If you need to use a different system jumbo MTU size for Layer 2 interfaces, see the “Configuring the System Jumbo MTU Size” section.

### SUMMARY STEPS

1. `configure terminal`
2. `interface ethernet slot/port`
3. `[switchport | no switchport]`
4. `mtu size`
5. `show interface ethernet slot/port`
6. `exit`
7. `copy running-config startup-config`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>configure terminal</code>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>switch# configure terminal</code>&lt;br&gt;<code>switch(config)#</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>interface ethernet slot/port</code>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>switch(config)# interface ethernet 3/1</code>&lt;br&gt;<code>switch(config-if)#</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>`[switchport</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><code>mtu size</code>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>switch(config-if)# mtu 9216</code>&lt;br&gt;<code>switch(config-if)#</code></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><code>show interface ethernet slot/port</code>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>switch(config)# show interface ethernet 2/1</code></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><code>exit</code>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>switch(config-if)# exit</code>&lt;br&gt;<code>switch(config)#</code></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><code>copy running-config startup-config</code>&lt;br&gt;<strong>Example:</strong></td>
</tr>
</tbody>
</table>
Configuring the System Jumbo MTU Size

You can configure the system jumbo MTU size, which can be used to specify the MTU size for Layer 2 interfaces. You can specify an even number between 1500 and 9216. If you do not configure the system jumbo MTU size, it defaults to 9216 bytes.

Note
To configure jumbo frames for FEX modules, configure the FEX fabric port-channel interface with the required MTU size for the FEX module.

SUMMARY STEPS

1. configure terminal
2. system jumbomtu size
3. show running-config all
4. interface type slot/port
5. interface type
6. mtu size
7. exit
8. copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 2**                 |                                                   |
| system jumbomtu size       | Specifies the system jumbo MTU size. Use an even number between 1500 and 9216. |
| Example:                   |                                                   |
### Configuring the System Jumbo MTU Size

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>switch(config)# system jumbomtu 8000</code>&lt;br&gt;<code>switch(config)#</code></td>
<td>In general accepted practice, a jumbo frame is considered to have an MTU size greater than 9000 bytes.</td>
</tr>
</tbody>
</table>

**Note**

- **Step 3**

  **show running-config all**
  
  **Example:**
  
  `switch(config)# show running-config all | include jumbomtu`

  (Optional) Displays the current operating configuration, which includes the system jumbo MTU size.

- **Step 4**

  **interface type slot/port**
  
  **Example:**
  
  `switch(config)# interface ethernet 2/1<br>`
  `switch(config-if)#`

  Specifies an interface to configure and enters interface configuration mode.

- **Step 5**

  **interface type**
  
  **Example:**
  
  `switch(config-if)# interface mgmt0<br>`
  `switch(config-if)#`

  Specifies the management interface to configure.

- **Step 6**

  **mtu size**
  
  **Example:**
  
  `switch(config-if)# mtu 1500<br>`
  `switch(config-if)#`

  For a Layer 2 interface, specifies either the default MTU size (1500) or the system jumbo MTU size that you specified earlier.<br>
  For a Layer 3 interface, specifies any even size between 576 and 9216.

- **Step 7**

  **exit**
  
  **Example:**
  
  `switch(config-if)# exit<br>`
  `switch(config)#`

  Exits the interface mode.

- **Step 8**

  **copy running-config startup-config**
  
  **Example:**
  
  `switch(config)# copy running-config startup-config`

  (Optional) Copies the running configuration to the startup configuration.

---

**Example**

This example shows how to configure the system jumbo MTU as 8000 bytes and how to change the MTU specification for an interface that was configured with the previous jumbo MTU size:

```
switch# configure terminal
switch(config)# system jumbomtu 8000
switch(config)# show running-config
switch(config)# interface ethernet 2/2
switch(config-if)# switchport
switch(config-if)# mtu 1500
switch(config-if)#
```
Configuring the Bandwidth

You can configure the bandwidth for Ethernet interfaces. The physical layer uses an unchangeable bandwidth of 1G, 10G, or 40G, but you can configure a value of 1 to 100,000,000 KB for Level 3 protocols.

SUMMARY STEPS

1. `configure terminal`
2. `interface ethernet slot/port`
3. `bandwidth kbps`
4. `show interface ethernet slot/port`
5. `exit`
6. `copy running-config startup-config`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>switch# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Specifies an Ethernet interface to configure, and enters interface configuration mode.</td>
</tr>
<tr>
<td><code>interface ethernet slot/port</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>switch(config)# interface ethernet 3/1</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config-if)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Specifies the bandwidth as an informational-only value between 1 and 100,000,000.</td>
</tr>
<tr>
<td><code>bandwidth kbps</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>switch(config-if)# bandwidth 1000000</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config-if)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>(Optional) Displays the interface status, which includes the bandwidth value.</td>
</tr>
<tr>
<td><code>show interface ethernet slot/port</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>switch(config)# show interface ethernet 2/1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Exits the interface mode.</td>
</tr>
<tr>
<td><code>exit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>switch(config-if)# exit</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td><code>copy running-config startup-config</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>switch(config)# copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>
Example

This example shows how to configure an informational value of 1,000,000 Kb for the Ethernet slot 3, port 1 interface bandwidth parameter:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# bandwidth 1000000
switch(config-if)#
```

Configuring the Throughput Delay

You can configure the interface throughput delay for Ethernet interfaces. The actual delay time does not change, but you can set an informational value between 1 and 16777215, where the value represents the number of tens of microseconds.

**SUMMARY STEPS**

1. configure terminal
2. interface ethernet slot/port
3. delay value
4. show interface ethernet slot/port
5. exit
6. copy running-config startup-config

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>interface ethernet slot/port</td>
<td>Specifies an Ethernet interface to configure, and enters interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# interface ethernet 3/1</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td>delay value</td>
<td>Specifies the delay time in tens of microseconds. You can set an informational value range between 1 and 16777215 tens of microseconds.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)# delay 10000</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td>show interface ethernet slot/port</td>
<td>(Optional) Displays the interface status, which includes the throughput-delay time.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# show interface ethernet 3/1</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)#</td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong> exit</td>
<td>Exits the interface mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>switch(config-if)# exit</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 6** copy running-config startup-config | (Optional) Copies the running configuration to the startup configuration. |
| **Example:**     |         |
| switch(config)# copy running-config startup-config |         |

### Example

This example shows how to configure the throughput-delay time so that one interface is preferred over another. A lower delay value is preferred over a higher value. In this example, Ethernet 7/48 is preferred over 7/47. The default delay for 7/48 is less than the configured value on 7/47, which is set for the highest value (16777215):

```
switch# configure terminal
switch(config)# interface ethernet 7/47
switch(config-if)# delay 16777215
switch(config-if)# ip address 192.168.10.1/24
switch(config-if)# ip router eigrp 10
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)# interface ethernet 7/48
switch(config-if)# ip address 192.168.11.1/24
switch(config-if)# ip router eigrp 10
switch(config-if)# no shutdown
switch(config-if)#
```

### Note

You must first ensure the EIGRP feature is enabled by running the `feature eigrp` command.

---

### Shutting Down and Activating the Interface

You can shut down and restart Ethernet or management interfaces. When you shut down interfaces, they become disabled and all monitoring displays show them as being down. This information is communicated to other network servers through all dynamic routing protocols. When the interfaces are shut down, the interface is not included in any routing updates. To activate the interface, you must restart the device.

#### SUMMARY STEPS

1. `configure terminal`
2. `interface interface`
3. `shutdown`
4. `show interface interface`
5. `no shutdown`
6. `show interface interface`
7. `exit`
8. `copy running-config startup-config`

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Step 1 | `configure terminal`  
Example:  
switch# configure terminal  
switch(config)# | Enters global configuration mode. |
| Step 2 | `interface interface`  
Example:  
switch(config)# interface ethernet 2/1  
switch(config)# interface mgmt0  
switch(config)# interface ethernet 2/1 | Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use `ethernet slot/port`. For the management interface, use `mgmt0`.  
Examples:  
• The 1st example shows how to specify the slot 2, port 1 Ethernet interface.  
• The 2nd example shows how to specify the management interface. |
| Step 3 | `shutdown`  
Example:  
switch(config-if)# shutdown  
switch(config-if)# | Disables the interface. |
| Step 4 | `show interface interface`  
Example:  
switch(config-if)# show interface ethernet 2/1  
switch(config-if)# | (Optional) Displays the interface status, which includes the administrative status. |
| Step 5 | `no shutdown`  
Example:  
switch(config-if)# no shutdown  
switch(config-if)# | Reenables the interface. |
| Step 6 | `show interface interface`  
Example:  
switch(config-if)# show interface ethernet 2/1  
switch(config-if)# | (Optional) Displays the interface status, which includes the administrative status. |
| Step 7 | `exit`  
Example:  
switch(config-if)# exit  
switch(config)# | Exits the interface mode. |
### Configuring the UDLD Mode

You can configure normal unidirectional link detection (UDLD) modes for Ethernet interfaces on devices configured to run UDLD.

Before you can enable a UDLD mode for an interface, you must make sure that UDLD is already enabled on the device that includes the interface. UDLD must also be enabled on the other linked interface and its device.

**Note**

If the interface is a copper port, you must use the command `enable UDLD` to enable the UDLD. If the interface is a fiber port you need not explicitly enable UDLD on the interface. However if you attempt to enable UDLD on a fiber port using the `enable UDLD` command, you may get an error message indicating that is not a valid command.

The following table lists CLI details to enable and disable UDLD on different interfaces

**Table 5: CLI Details to Enable or Disable UDLD on Different Interfaces**

<table>
<thead>
<tr>
<th>Description</th>
<th>Fiber port</th>
<th>Copper or Nonfiber port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default setting</td>
<td>Enabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>Enable UDLD command</td>
<td>no udlld disable</td>
<td>udlld enable</td>
</tr>
<tr>
<td>Disable UDLD command</td>
<td>udlld disable</td>
<td>no udlld enable</td>
</tr>
</tbody>
</table>

### Configuring Basic Interface Parameters

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 8</strong></td>
<td><em>(Optional)</em> Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td><code>copy running-config startup-config</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch(config)# copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>

**Example**

This example shows how to change the administrative status for Ethernet port 3/1 from disabled to enabled:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# shutdown
switch(config-if)# no shutdown
```

**SUMMARY STEPS**

1. `configure terminal`
2. `[no] feature udld`
3. `udld message-time seconds`
4. `udld aggressive`
5. `interface ethernet slot/port`
6. `udld [enable | disable]`
7. `show udld [ethernet slot/port | global | neighbors]`
8. `exit`
9. `copy running-config startup-config`

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: switch# configure terminal switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>no</code> feature udld</td>
<td>Enables/Disables UDLD for the device.</td>
</tr>
<tr>
<td>Example: switch(config)# feature udld switch(config)#</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)# no feature udld switch(config)#</td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>udld message-time seconds</code></td>
<td>(Optional) Specifies the interval between sending UDLD messages. The range is from 7 to 90 seconds, and the default is 15 seconds.</td>
</tr>
<tr>
<td>Example: switch(config)# udld message-time 30 switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>udld aggressive</code></td>
<td>Optional) Specifies UDLD mode to be aggressive.</td>
</tr>
<tr>
<td>Example: switch(config)# udld aggressive switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>interface ethernet slot/port</code></td>
<td>(Optional) Specifies an interface to configure, and enters interface configuration mode.</td>
</tr>
<tr>
<td>Example: switch(config)# interface ethernet 3/1 switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> `udld [enable</td>
<td>disable]`</td>
</tr>
<tr>
<td>Example: switch(config-if)# udld enable switch(config-if)#</td>
<td>To enable UDLD on copper ports, enter the <code>udld enable</code> command. To enable UDLD on fiber ports, enter the <code>no udld disable</code> command.</td>
</tr>
</tbody>
</table>
## Configuring Debounce Timers

You can enable the debounce timer for Ethernet ports by specifying a debounce time (in milliseconds) or disable the timer by specifying a debounce time of 0.
The link state of 10G and 100G ports may change repeatedly when connected to service provider network. As a part of link reset or break-link functionality, it is expected that the Tx power light on the SFP to change to N/A state, at an event of link state change.

However, to prevent this behavior during the link state change, you may increase the link debounce timer to start from 500ms and increase it in 500ms intervals until the link stabilizes. On the DWDM, UVN, and WAN network, it is recommended to disable automatic link suspension (ALS) whenever possible. ALS suspends the link on the WAN when the Nexus turn off the link.

Note

The link debounce time and link debounce link-up time commands can only be applied to a physical Ethernet interface.

Use the show interface debounce command to display the debounce times for all Ethernet ports.

The link debounce time command is not supported on 10G and 40G ports on the Cisco Nexus 93300YC-FX and Cisco Nexus 9336C-FX switches.

The link debounce time command is supported on 1G, 10G, 40G, 25G and 100G SFP/QSFP ports on the Cisco Nexus 9000 series switches.


SUMMARY STEPS

1. configure terminal
2. interface ethernet slot/port
3. link debounce time time
4. link debounce link-up time

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td></td>
</tr>
<tr>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
| switch# configure terminal
switch(config)#        |                                              |
| Step 2                  |                                              |
| interface ethernet slot/port | Specifies an Ethernet interface to configure, and enters interface configuration mode. |
| Example:                |                                              |
| switch(config)# interface ethernet 3/1
switch(config-if)#     |                                              |
| Step 3                  |                                              |
| link debounce time time | Enables the debounce timer for the specified time (1 to 5000 milliseconds). |
| Example:                | If you specify 0 milliseconds, the debounce timer is disabled. |
| switch(config-if)# link debounce time 1000
switch(config-if)#     |                                              |
### Configuring Port Profiles

You can apply several configuration parameters to a range of interfaces simultaneously. All the interfaces in the range must be the same type. You can also inherit the configurations from one port profile into another port profile. The system supports four levels of inheritance.

### Creating a Port Profile

You can create a port profile on the device. Each port profile must have a unique name across types and the network.
Port profile names can include only the following characters:

- a-z
- A-Z
- 0-9
- No special characters are allowed, except for the following:
  - .
  - -
  - _

SUMMARY STEPS

1. configure terminal
2. port-profile [type {ethernet | interface-vlan | port-channel}] name
3. exit
4. (Optional) show port-profile
5. (Optional) copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td>Step 2 port-profile [type {ethernet</td>
<td>interface-vlan</td>
</tr>
<tr>
<td>Step 3 exit</td>
<td>Exits the port-profile configuration mode.</td>
</tr>
<tr>
<td>Step 4 (Optional) show port-profile</td>
<td>Displays the port-profile configuration.</td>
</tr>
<tr>
<td>Step 5 (Optional) copy running-config startup-config</td>
<td>Copies the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>

Example

This example shows how to create a port profile named test for ethernet interfaces:

```
switch# configure terminal
switch(config)# port-profile type ethernet test
switch(config-ppm)#
```
Entering Port-Profile Configuration Mode and Modifying a Port Profile

You can enter the port-profile configuration mode and modify a port profile. To modify the port profile, you must be in the port-profile configuration mode.

SUMMARY STEPS

1. configure terminal
2. port-profile [type {ethernet | interface-vlan | port-channel}] name
3. exit
4. (Optional) show port-profile
5. (Optional) copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td>Step 2 port-profile [type {ethernet</td>
<td>interface-vlan</td>
</tr>
<tr>
<td>Step 3 exit</td>
<td>Exits the port-profile configuration mode.</td>
</tr>
<tr>
<td>Step 4 (Optional) show port-profile</td>
<td>Displays the port-profile configuration.</td>
</tr>
<tr>
<td>Step 5 (Optional) copy running-config startup-config</td>
<td>Copies the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>

Example

This example shows how to enter the port-profile configuration mode for the specified port profile and bring all the interfaces administratively up:

```
switch# configure terminal
switch(config)# port-profile type ethernet test
switch(config-ppm)# no shutdown
switch(config-ppm)#
```

Assigning a Port Profile to a Range of Interfaces

You can assign a port profile to an interface or to a range of interfaces. All the interfaces must be the same type.

SUMMARY STEPS

1. configure terminal
2. interface [ethernet slot/port | interface-vlan vlan-id | port-channel number]
3. inherit port-profile name
4. exit
5. (Optional) show port-profile
6. (Optional) copy running-config startup-config

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td>Step 2 interface [ethernet slot/port</td>
<td>interface-vlan vlan-id</td>
</tr>
<tr>
<td>Step 3 inherit port-profile name</td>
<td>Assigns the specified port profile to the selected interfaces.</td>
</tr>
<tr>
<td>Step 4 exit</td>
<td>Exits the port-profile configuration mode.</td>
</tr>
<tr>
<td>Step 5 (Optional) show port-profile</td>
<td>Displays the port-profile configuration.</td>
</tr>
<tr>
<td>Step 6 (Optional) copy running-config startup-config</td>
<td>Copies the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>

### Example

This example shows how to assign the port profile named adam to Ethernet interfaces 7/3 to 7/5, 10/2, and 11/20 to 11/25:

```
switch# configure terminal
switch(config)# interface ethernet7/3-5, ethernet10/2, ethernet11/20-25
switch(config-if)# inherit port-profile adam
```

### Enabling a Specific Port Profile

To apply the port-profile configurations to the interfaces, you must enable the specific port profile. You can configure and inherit a port profile onto a range of interfaces before you enable that port profile. You would then enable that port profile for the configurations to take effect on the specified interfaces.

If you inherit one or more port profiles onto an original port profile, only the last inherited port profile must be enabled; the system assumes that the underlying port profiles are enabled.

You must be in the port-profile configuration mode to enable or disable port profiles.

### SUMMARY STEPS

1. configure terminal
2. port-profile [type {ethernet | interface-vlan | port-channel}] name
3. state enabled
4. exit
5. (Optional) show port-profile
6. (Optional) copy running-config startup-config
Inheriting a Port Profile

You can inherit a port profile onto an existing port profile. The system supports four levels of inheritance.

SUMMARY STEPS

1. configure terminal
2. port-profile name
3. inherit port-profile name
4. exit
5. (Optional) show port-profile
6. (Optional) copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td>Step 2 port-profile name</td>
<td>Enters the port-profile configuration mode for the specified port profile.</td>
</tr>
<tr>
<td>Step 3 inherit port-profile name</td>
<td>Inherits another port profile onto the existing one. The original port profile assumes all the configurations of the inherited port profile.</td>
</tr>
</tbody>
</table>
### Command or Action | Purpose
--- | ---
**Step 4** | exit
**Step 5** | (Optional) *show port-profile*
**Step 6** | (Optional) *copy running-config startup-config*

**Example**

This example shows how to inherit the port profile named adam onto the port profile named test:

```
switch# configure terminal
switch(config)# port-profile test
switch(config-ppm)# inherit port-profile adam
switch(config-ppm)#
```

### Removing a Port Profile from a Range of Interfaces

You can remove a port profile from some or all of the interfaces to which you have applied the profile. You do this configuration in the interfaces configuration mode.

**SUMMARY STEPS**

1. `configure terminal`
2. `interface [ethernet slot/port | interface-vlan vlan-id | port-channel number]`
3. `no inherit port-profile name`
4. `exit`
5. (Optional) `show port-profile`
6. (Optional) `copy running-config startup-config`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>configure terminal</code> Enters the global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>`interface [ethernet slot/port</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>no inherit port-profile name</code> Un-assigns the specified port profile to the selected interfaces.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><code>exit</code> Exits the port-profile configuration mode.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>(Optional) <code>show port-profile</code> Displays the port-profile configuration.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>(Optional) <code>copy running-config startup-config</code> Copies the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>
Example
This example shows how to unassign the port profile named adam to Ethernet interfaces 7/3 to 7/5, 10/2, and 11/20 to 11/25:

```
switch# configure terminal
switch(config)# interface ethernet 7/3-5, 10/2, 11/20-25
switch(config-if)# no inherit port-profile adam
switch(config-if)#
```

Removing an Inherited Port Profile

You can remove an inherited port profile. You do this configuration in the port-profile mode.

**SUMMARY STEPS**

1. `configure terminal`
2. `port-profile name`
3. `no inherit port-profile name`
4. `exit`
5. (Optional) `show port-profile`
6. (Optional) `copy running-config startup-config`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>configure terminal</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>port-profile name</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>no inherit port-profile name</code></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><code>exit</code></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>(Optional) <code>show port-profile</code></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>(Optional) <code>copy running-config startup-config</code></td>
</tr>
</tbody>
</table>

Example
This example shows how to remove the inherited port profile named adam from the port profile named test:

```
switch# configure terminal
switch(config)# port-profile test
switch(config-ppm)# no inherit port-profile adam
switch(config-ppm)#
```
Configuring 25G Autonegotiation

Autonegotiation allows devices to advertise enhanced modes of operation it possesses via the link segment and to detect corresponding enhanced operational modes that the other devices may be advertising. Autonegotiation provides the means to exchange information between two devices that share a link segment and to automatically configure both devices to take maximum advantage of their abilities.

Guidelines and Limitations for 25G Autonegotiation

- Autonegotiation is not supported on Cisco Nexus N9K-C92300YC switch.
- Autonegotiation is not supported on 25G breakout ports.

FEC selection with 25G Autonegotiation

Table 6: FEC Selection with 25G Autonegotiation

<table>
<thead>
<tr>
<th>Hardware</th>
<th>1m</th>
<th>2m</th>
<th>3m</th>
<th>5m</th>
</tr>
</thead>
<tbody>
<tr>
<td>N9K-C93240YC-FX2</td>
<td>No FEC</td>
<td>No FEC</td>
<td>FC-FEC</td>
<td>RS-IEEE</td>
</tr>
<tr>
<td>N9K-C93180YC-FX</td>
<td>No FEC</td>
<td>No FEC</td>
<td>FC-FEC</td>
<td>RS-IEEE</td>
</tr>
<tr>
<td>N9K-C93180YC-EX</td>
<td>No FEC</td>
<td>No FEC</td>
<td>FC-FEC</td>
<td>FC-FEC</td>
</tr>
<tr>
<td>N9K-X97160YC-EX</td>
<td>No FEC</td>
<td>No FEC</td>
<td>FC-FEC</td>
<td>FC-FEC</td>
</tr>
</tbody>
</table>

Note

25G autonegotiation is not supported on Cisco Nexus N9K-C92300YC switch.

Enabling Autonegotiation

You can enable autonegotiation using the `negotiate auto` command. To enable autonegotiation, follow these steps:

SUMMARY STEPS

1. `configure terminal`
2. `interface ethernet port number`
3. `negotiate auto port speed`
Disabling Autonegotiation

You can disable autonegotiation using the `no negotiate auto` command. To disable autonegotiation, follow these steps:

1. `configure terminal`
2. `interface ethernet port number`
3. `no negotiate auto port speed`

## SUMMARY STEPS

1. `configure terminal`
2. `interface ethernet port number`
3. `no negotiate auto port speed`

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface ethernet port number</td>
<td>Selects the interface and enters interface mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# int e1/7</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> negotiate auto port speed</td>
<td>Enables autonegotiation on the selected interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td>negotiate auto 25000</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)#</td>
<td></td>
</tr>
</tbody>
</table>

This example shows how to enable autonegotiation on a specified interface:

**Example**

```
switch# sh int e1/7 st
--------------------------------------------------------------------------------
Port   Name   Status    Vlan Duplex Speed    Type
--------------------------------------------------------------------------------
Eth1/7 -- connected routed full 25G SFP-H25GB-CU1M
switch# conf
switch(config)# int e1/7
switch(config-if)# negotiate auto 25000
```

### Disabling Autonegotiation

You can disable autonegotiation using the `no negotiate auto` command. To disable autonegotiation, follow these steps:
### Configuring Basic Interface Parameters

#### Step 2

**interface ethernet port number**

Selects the interface and enters interface mode.

**Example:**

```
switch# int e1/7
switch(config-if)#
```

#### Step 3

**no negotiate auto port speed**

Disables autonegotiation on the selected interface.

**Example:**

```
switch(config-if)# no negotiate auto 25000
switch(config-if)#
```

**Note:** You must apply this command on interfaces at both sides of the link.

---

This example shows how to disable autonegotiation on a specified interface.

**Example**

```
switch# sh int e1/7 st
```

```
+---------------------------------------------------------------------+
<table>
<thead>
<tr>
<th>Port</th>
<th>Name</th>
<th>Status</th>
<th>Vlan</th>
<th>Duplex</th>
<th>Speed</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eth1/7</td>
<td>--</td>
<td>connected routed</td>
<td>full</td>
<td>25G</td>
<td>SFP-H25GB-CU1M</td>
<td></td>
</tr>
</tbody>
</table>
```

```
switch# conf
switch(config)# int e1/7
switch(config-if)# no negotiate auto 25000
```

---

### Verifying the Basic Interface Parameters

You can verify the basic interface parameters by displaying their values. You can also clear the counters listed when you display the parameter values.

To display basic interface configuration information, perform one of the following tasks:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>show cdp all</strong></td>
<td>Displays the CDP status.</td>
</tr>
<tr>
<td><strong>show interface interface</strong></td>
<td>Displays the configured states of one or all interfaces.</td>
</tr>
<tr>
<td><strong>show interface brief</strong></td>
<td>Displays a table of interface states.</td>
</tr>
<tr>
<td><strong>show interface status err-disabled</strong></td>
<td>Displays information about error-disabled interfaces.</td>
</tr>
<tr>
<td><strong>show udlld interface</strong></td>
<td>Displays the UDLD status for the current interface or all interfaces.</td>
</tr>
<tr>
<td><strong>show udlld global</strong></td>
<td>Displays the UDLD status for the current device.</td>
</tr>
</tbody>
</table>
Monitoring the Interface Counters

You can display and clear interface counters using Cisco NX-OS.

Displaying Interface Statistics

You can set up to three sampling intervals for statistics collections on interfaces.

SUMMARY STEPS

1. `configure terminal`
2. `interface ether slot/port`
3. `load-interval counters [1 | 2 | 3] seconds`
4. `show interface interface`
5. `exit`
6. `copy running-config startup-config`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
</tbody>
</table>
| **Example:**      | switch# configure terminal  
switch(config)# |
| **Step 2**        | interface ether *slot/port* |
| **Example:**      | switch(config)# interface ether 4/1  
switch(config)# |
| **Step 3**        | load-interval counters [1 | 2 | 3] seconds |
| **Example:**      | switch(config)# load-interval counters 1 100  
switch(config)# |
| **Step 4**        | show interface *interface* |
| **Example:**      | switch(config)# show interface ethernet 2/2  
switch# |
| **Step 5**        | exit |
| **Example:**      | |
### Clearing Interface Counters

You can clear the Ethernet and management interface counters by using the `clear counters interface` command. You can perform this task from the configuration mode or interface configuration mode.

#### SUMMARY STEPS

1. `clear counters interface [all | ethernet slot/port | loopback number | mgmt number | port channel channel-number]`
2. `show interface interface`
3. `show interface [ethernet slot/port | port channel channel-number] counters`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong>&lt;br&gt;clear counters interface [all</td>
<td>ethernet slot/port</td>
</tr>
<tr>
<td><strong>Step 2</strong>&lt;br&gt;show interface interface&lt;br&gt;Example: switch# show interface ethernet 2/1 switch#</td>
<td>(Optional) Displays the interface status.</td>
</tr>
<tr>
<td><strong>Step 3</strong>&lt;br&gt;show interface [ethernet slot/port</td>
<td>port channel channel-number] counters&lt;br&gt;Example: switch# show interface ethernet 2/1 switch#</td>
</tr>
</tbody>
</table>
### Example

This example shows how to clear the counters on Ethernet port 5/5:

```
switch# clear counters interface ethernet 5/5
switch#
```

### Configuration Example for QSA

For a Cisco Nexus 9396PX:

- Using the default configuration on port 2/1, all the QSFPs in port group 2/1-6 are brought up with a speed of 40G. If there are any QSA modules in port group 2/1-6, they are error disabled.

- Using the `speed-group [10000 | 40000]` command to configure port 2/7, all the QSAs in port group 2/7-12 are brought up with a speed of 10G or 40G. If there are any QSFP modules in port group 2/7-12, they are error disabled.

This example shows how to configure QSA for the first port in the speed group for a Cisco Nexus 9396PX:

```
switch# conf t
switch(config)# interface ethernet 2/7
switch(config-if)# speed-group 10000
```
Chapter 4

Configuring Layer 2 Interfaces

- Information About Access and Trunk Interfaces, on page 55
- Licensing Requirements for Layer 2 Port Modes, on page 61
- Prerequisites for Layer 2 Interfaces, on page 62
- Guidelines and Limitations for Layer 2 Interfaces, on page 62
- Default Settings for Layer 2 Interfaces, on page 65
- Configuring Access and Trunk Interfaces, on page 65
- Verifying the Interface Configuration, on page 87
- Monitoring the Layer 2 Interfaces, on page 87
- Configuration Examples for Access and Trunk Ports, on page 88
- Related Documents, on page 88

Information About Access and Trunk Interfaces

- Note See the Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide for complete information on high-availability features.

- Note The device supports only IEEE 802.1Q-type VLAN trunk encapsulation.

About Access and Trunk Interfaces

A Layer 2 port can be configured as an access or a trunk port as follows:

- An access port can have only one VLAN configured on that port; it can carry traffic for only one VLAN.
- A trunk port can have two or more VLANs configured on that port; it can carry traffic for several VLANs simultaneously.

By default, all the ports on Cisco Nexus 9300-EX switches are Layer 3 ports and all the ports on Cisco Nexus 9300 switches are Layer 2 ports.
You can make all ports Layer 2 ports using the setup script or by entering the `system default switchport` command. See the Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide for information about using the setup script. To configure the port as a Layer 2 port using the CLI, use the `switchport` command.

All ports in the same trunk must be in the same VDC, and trunk ports cannot carry VLANs from different VDCs.

The following figure shows how you can use trunk ports in the network. The trunk port carries traffic for two or more VLANs.

*Figure 2: Trunk and Access Ports and VLAN Traffic*

Note

See the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for information about VLANs.

In order to correctly deliver the traffic on a trunk port with several VLANs, the device uses the IEEE 802.1Q encapsulation, or tagging, method (see the “IEEE 802.1Q Encapsulation” section for more information).

Note

See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for information about subinterfaces on Layer 3 interfaces.

To optimize the performance on access ports, you can configure the port as a host port. Once the port is configured as a host port, it is automatically set as an access port, and channel grouping is disabled. Use the host designation to decrease the time that it takes the designated port to begin to forward packets.

Only an end station can be set as a host port; you will receive an error message if you attempt to configure other ports as hosts.

If an access port receives a packet with an 802.1Q tag in the header other than the access VLAN value, that port drops the packet without learning its MAC source address.

A Layer 2 interface can function as either an access port or a trunk port; it cannot function as both port types simultaneously.

When you change a Layer 2 interface back to a Layer 3 interface, that interface loses all the Layer 2 configuration and resumes the default VLAN configurations.
IEEE 802.1Q Encapsulation

Note

For information about VLANs, see the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide.

A trunk is a point-to-point link between the switch and another networking device. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across an entire network.

To correctly deliver the traffic on a trunk port with several VLANs, the device uses the IEEE 802.1Q encapsulation, or tagging, method that uses a tag that is inserted into the frame header. This tag carries information about the specific VLAN to which the frame and packet belong. This method allows packets that are encapsulated for several different VLANs to traverse the same port and maintain traffic separation between the VLANs. Also, the encapsulated VLAN tag allows the trunk to move traffic end-to-end through the network on the same VLAN.

Figure 3: Header Without and With 802.1Q Tag

<table>
<thead>
<tr>
<th>Preamble (7-bytes)</th>
<th>Start Frame Delimiter (1-byte)</th>
<th>Dest. MAC Address (6-bytes)</th>
<th>Source MAC Address (6-bytes)</th>
<th>Length/Type (2-bytes)</th>
<th>MAC Client Data (0-n bytes)</th>
<th>Pad (0-p bytes)</th>
<th>Frame Check Sequence (4-bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3 bits = User Priority field
1 bit = Canonical Format Identifier (CFI)
12 bits = VLAN Identifier (VLAN ID)

Access VLANs

When you configure a port in access mode, you can specify which VLAN will carry the traffic for that interface. If you do not configure the VLAN for a port in access mode, or an access port, the interface carries traffic for the default VLAN (VLAN1).

You can change the access port membership in a VLAN by specifying the new VLAN. You must create the VLAN before you can assign it as an access VLAN for an access port. If you change the access VLAN on an access port to a VLAN that is not yet created, the system shuts that access port down.
If an access port receives a packet with an 802.1Q tag in the header other than the access VLAN value, that port drops the packet without learning its MAC source address.

Native VLAN IDs for Trunk Ports

A trunk port can carry nontagged packets simultaneously with the 802.1Q tagged packets. When you assign a default port VLAN ID to the trunk port, all untagged traffic travels on the default port VLAN ID for the trunk port, and all untagged traffic is assumed to belong to this VLAN. This VLAN is referred to as the native VLAN ID for a trunk port. That is, the native VLAN ID is the VLAN that carries untagged traffic on trunk ports.

**Note**
Native VLAN ID numbers must match on both ends of the trunk.

The trunk port sends an egressing packet with a VLAN that is equal to the default port VLAN ID as untagged; all the other egressing packets are tagged by the trunk port. If you do not configure a native VLAN ID, the trunk port uses the default VLAN.

**Note**
You cannot use a Fibre Channel over Ethernet (FCoE) VLAN as a native VLAN for an Ethernet trunk switchport.

Tagging Native VLAN Traffic

The Cisco software supports the IEEE 802.1Q standard on trunk ports. In order to pass untagged traffic through the trunk ports, you must create a VLAN that does not tag any packets (or you can use the default VLAN). Untagged packets can pass through trunk ports and access ports.

However, all packets that enter the device with an 802.1Q tag that matches the value of the native VLAN on the trunk are stripped of any tagging and egress the trunk port as untagged packets. This situation can cause problems because you may want to retain the tagging on packets on the native VLAN for the trunk port.

You can configure the device to drop all untagged packets on the trunk ports and to retain the tagging of packets entering the device with 802.1Q values that are equal to that of the native VLAN ID. All control traffic still passes on the native VLAN. This configuration is global; trunk ports on the device either do or do not retain the tagging for the native VLAN.

Allowed VLANs

By default, a trunk port sends traffic to and receives traffic from all VLANs. All VLAN IDs are allowed on each trunk. However, you can remove VLANs from this inclusive list to prevent traffic from the specified VLANs from passing over the trunk. Later, you can add any specific VLANs that you may want the trunk to carry traffic for back to the list.

To partition the Spanning Tree Protocol (STP) topology for the default VLAN, you can remove VLAN1 from the list of allowed VLANs. Otherwise, VLAN1, which is enabled on all ports by default, will have a very big STP topology, which can result in problems during STP convergence. When you remove VLAN1, all data traffic for VLAN1 on this port is blocked, but the control traffic continues to move on the port.
See the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for more information about STP.

You can change the block of VLANs reserved for internal use. See the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for more information about changing the reserved VLANs.

**Enabling 4K VLAN Configurations with switchport isolated**

The switchport isolated feature allows disabling STP on an interface. Using this feature allows a maximum number of 4K VLAN * 48 virtual ports. Configuring the switchport isolated feature places all 4K VLANS in forwarding state for that port. (Removing a VLAN does not bring down the logical port.)

The feature is supported on MSTP modes. It is also supported on physical interfaces and port-channels (including vPC).

The switchport isolated feature supports a maximum of 48 ports with 4K VLANS in MSTP modes.

In vPC configurations, the Type-1 consistency check is performed between vPC peers. If the check result is inconsistent, the secondary vPC is brought down while the primary continues to stay up.

Spanning-tree is not notified during logical bring up/bring down when using the switchport isolated feature.

**Default Interfaces**

You can use the default interface feature to clear the configured parameters for both physical and logical interfaces such as the Ethernet, loopback, VLAN network, tunnel, and the port-channel interface.

A maximum of eight ports can be selected for the default interface. The default interfaces feature is not supported for management interfaces because the device could go to an unreachable state.

**Switch Virtual Interface and Autostate Behavior**

In Cisco NX-OS, a switch virtual interface (SVI) represents a logical interface between the bridging function and the routing function of a VLAN in the device.

The operational state of this interface is governed by the state of the various ports in its corresponding VLAN. An SVI interface on a VLAN comes up when at least one port in that VLAN is in the Spanning Tree Protocol (STP) forwarding state. Similarly, this interface goes down when the last STP forwarding port goes down or goes to another STP state.
SVI Autostate Exclude

Typically, when a VLAN interface has multiple ports in the VLAN, the SVI goes to the down state when all the ports in the VLAN go down. You can use the SVI autostate exclude feature to exclude specific ports and port channels while defining the status of the SVI (up or down) even if it belongs to the same VLAN. For example, even if the excluded port or port channel is in the up state and other ports are in the down state in the VLAN, the SVI state is changed to down.

Note

You can use the SVI autostate exclude feature only for switched physical Ethernet ports and port channels.

SVI Autostate Disable

You can configure the autostate disable feature to keep an SVI up even if no interface is up in the corresponding VLAN. You can configure this feature for the system (for all SVIs) or for an individual SVI.

High Availability

The software supports high availability for Layer 2 ports.

Note

See the Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide for complete information about high availability features.

Virtualization Support

All ports in the same trunk must be in the same VDC, and trunk ports cannot carry VLANs from different VDCs.

Counter Values

See the following information on the configuration, packet size, incremented counter values, and traffic.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Packet Size</th>
<th>Incremented Counters</th>
<th>Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>L2 port – without any MTU configuration</td>
<td>6400 and 10000</td>
<td>Jumbo, giant, and input error</td>
<td>Dropped</td>
</tr>
<tr>
<td>L2 port – with jumbo MTU 9216 in network-qos config</td>
<td>6400</td>
<td>Jumbo</td>
<td>Forwarded</td>
</tr>
<tr>
<td>L2 port – with jumbo MTU 9216 in network-qos config</td>
<td>10000</td>
<td>Jumbo, giant, and input error</td>
<td>Dropped</td>
</tr>
</tbody>
</table>
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Packet Size</th>
<th>Incremented Counters</th>
<th>Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 3 port with default Layer 3 MTU and jumbo MTU 9216 in network-qos configuration</td>
<td>6400</td>
<td>Jumbo</td>
<td>Packets are punted to the CPU (subjected to CoPP configs), get fragmented, and then they are forwarded by the software.</td>
</tr>
<tr>
<td>Layer 3 port with default Layer 3 MTU and jumbo MTU 9216 in network-qos configuration</td>
<td>6400</td>
<td>Jumbo</td>
<td>Packets are punted to the CPU (subjected to CoPP configs), get fragmented, and then they are forwarded by the software.</td>
</tr>
<tr>
<td>Layer 3 port with default Layer 3 MTU and jumbo MTU 9216 in network-qos configuration</td>
<td>10000</td>
<td>Jumbo, giant, and input error</td>
<td>Dropped</td>
</tr>
<tr>
<td>Layer 3 port with jumbo Layer 3 MTU and jumbo MTU 9216 in network-qos configuration</td>
<td>6400</td>
<td>Jumbo</td>
<td>Forwarded without any fragmentation.</td>
</tr>
<tr>
<td>Layer 3 port with jumbo Layer 3 MTU and jumbo MTU 9216 in network-qos configuration</td>
<td>10000</td>
<td>Jumbo, giant, and input error</td>
<td>Dropped</td>
</tr>
<tr>
<td>Layer 3 port with jumbo Layer 3 MTU and default L2 MTU configuration</td>
<td>6400 and 10000</td>
<td>Jumbo, giant, and input error</td>
<td>Dropped</td>
</tr>
</tbody>
</table>

### Note

- Under 64 bytes packet with good CRC–The short frame counter increments.
- Under 64 bytes packet with bad CRC–The runts counter increments.
- Greater than 64 bytes packet with bad CRC–The CRC counter increments.

### Licensing Requirements for Layer 2 Port Modes

The following table shows the licensing requirements for this feature:
### Prerequisites for Layer 2 Interfaces

Layer 2 interfaces have the following prerequisites:

- You are logged onto the device.
- You must configure the port as a Layer 2 port before you can use the `switchport mode` command. By default, all ports on the device are Layer 3 ports. By default, all ports on the Cisco Nexus 9504 and Cisco Nexus 9508 devices are Layer 2 ports.

### Guidelines and Limitations for Layer 2 Interfaces

VLAN trunking has the following configuration guidelines and limitations:

- Cisco Nexus 9000 Series switches have the `vlan dot1q tag native` command enabled by default. This tags the native VLAN on the configured trunk ports. However, connected switches such as Catalyst 6500 or third-party switches, probably would not have a similar configuration enabled. This could result in unexpected behaviors. Therefore, it is recommended to have the `vlan dot1q tag native` command disabled in case the connected switch does not have it configured.

- Auto-negotiation is not supported on Cisco Nexus 9508 platform switches with N9K-X96136C-R, N9K-X96136C-RX, and N9K-X9636Q-R line cards.

- Beginning with Cisco NX-OS Release 9.2(1), the Cisco Nexus 9508 platform switches with N9K-X96136YC-R line cards support 1 Gigabit speed on all 48 ports. However, because the auto negotiation is not supported, 1000BASE-T SFPs links comes up even the cable is removed.


  **Note**  
  Auto negotiation is not supported on Cisco Nexus N9K-C92300YC switch

- `show` commands with the `internal` keyword are not supported.
- On Cisco Nexus 9300 platform switches, a unicast ARP request to SVI is flooded to the other ports within the VLAN.
- On Cisco Nexus 9300 platform switches, a unicast ARP request to SVI is flooded to the other ports within the VLAN.

<table>
<thead>
<tr>
<th>Product</th>
<th>License Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco NX-OS</td>
<td>Layer 2 port modes require no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you.</td>
</tr>
</tbody>
</table>
• ASE2 and ASE3 based Cisco Nexus 9000 Series switches acting as transit switches do not preserve the inner tag for double-tagged packets.

The following CLI is mandatory only on LSE based Cisco Nexus 9000 Series switches. For seamless packet forwarding and preservation of all VLAN tags on pure transit boxes in the SP cloud that have no Q-in-Q encapsulation or decapsulation requirement, configure the CLI command, `system dot1q-tunnel transit`. To remove the CLI, use `no system dot1q-tunnel transit` CLI command.

The caveats with the CLI that is executed on the switches are:

- L2 frames that egress out of the trunk ports are tagged even on the native VLAN on the port.
- Any other tunneling mechanism, for example, VXLAN and MPLS does not work with the CLI configured.

• A port can be either a Layer 2 or a Layer 3 interface; it cannot be both simultaneously.
• When you change a Layer 3 port to a Layer 2 port or a Layer 2 port to a Layer 3 port, all layer-dependent configuration is lost. When you change an access or trunk port to a Layer 3 port, all information about the access VLAN, native VLAN, allowed VLANs, and so forth, is lost.
• Do not connect devices with access links because access links may partition a VLAN.
• When connecting Cisco devices through an 802.1Q trunk, make sure that the native VLAN for an 802.1Q trunk is the same on both ends of the trunk link. If the native VLAN on one end of the trunk is different from the native VLAN on the other end, spanning tree loops might result.
• Disabling spanning tree on the native VLAN of an 802.1Q trunk without disabling spanning tree on every VLAN in the network can cause spanning tree loops. You must leave spanning tree enabled on the native VLAN of an 802.1Q trunk. If you cannot leave spanning tree enabled, you must disable spanning tree on every VLAN in the network. Make sure that your network has no physical loops before you disable spanning tree.

• When you connect two Cisco devices through 802.1Q trunks, the devices exchange spanning tree bridge protocol data units (BPDUs) on each VLAN allowed on the trunks. The BPDUs on the native VLAN of the trunk are sent untagged to the reserved IEEE 802.1D spanning tree multicast MAC address (01-80-C2-00-00-00). The BPDUs on all other VLANs on the trunk are sent tagged to the reserved Cisco Shared Spanning Tree (SSTP) multicast MAC address (01-00-0c-cc-cc-cd).

• Non-Cisco 802.1Q devices maintain only a single instance of spanning tree (the Mono Spanning Tree) that defines the spanning tree topology for all VLANs. When you connect a Cisco switch to a non-Cisco switch through an 802.1Q trunk, the Mono Spanning Tree of the non-Cisco switch and the native VLAN spanning tree of the Cisco switch combine to form a single spanning tree topology known as the Common Spanning Tree (CST).

• Because Cisco devices transmit BPDUs to the SSTP multicast MAC address on VLANs other than the native VLAN of the trunk, non-Cisco devices do not recognize these frames as BPDUs and flood them on all ports in the corresponding VLAN. Other Cisco devices connected to the non-Cisco 802.1Q cloud receive these flooded BPDUs. This BPDUs reception allows Cisco switches to maintain a per-VLAN spanning tree topology across a cloud of non-Cisco 802.1Q devices. The non-Cisco 802.1Q cloud that separates the Cisco devices is treated as a single broadcast segment between all devices connected to the non-Cisco 802.1Q cloud through 802.1Q trunks.

• Make certain that the native VLAN is the same on all of the 802.1Q trunks that connect the Cisco devices to the non-Cisco 802.1Q cloud.
• If you are connecting multiple Cisco devices to a non-Cisco 802.1Q cloud, all of the connections must be through 802.1Q trunks. You cannot connect Cisco devices to a non-Cisco 802.1Q cloud through access ports because doing so places the access port on the Cisco device into the spanning tree “port inconsistent” state and no traffic will pass through the port.

• You can group trunk ports into port-channel groups, but all trunks in the group must have the same configuration. When a group is first created, all ports follow the parameters set for the first port to be added to the group. If you change the configuration of one of these parameters, the device propagates that setting to all ports in the group, such as the allowed VLANs and the trunk status. For example, if one port in a port group ceases to be a trunk, all ports cease to be trunks.

• If you try to enable 802.1X on a trunk port, an error message appears, and 802.1X is not enabled. If you try to change the mode of an 802.1X-enabled port to trunk, the port mode is not changed.

• Only ingress unicast packet counters are supported for SVI counters.

• When MAC addresses are cleared on a VLAN with the clear mac address-table dynamic command, the dynamic ARP (Address Resolution Protocol) entries on that VLAN are refreshed.

• If a static ARP entry exists on the VLAN and no MAC address to port mapping is present, the supervisor may generate an ARP request to learn the MAC address. Upon learning the MAC address, the adjacency entry points to the correct physical port.

• Cisco NX-OS does not support transparent bridging between two VLANs when one of the SVIs is on the Cisco Nexus 9000 using the BIA MAC (burned-in MAC address). This occurs when the BIA MAC is shared between SVIs/VLANs. A MAC, different from the BIA MAC, can be configured under the SVI for transparent bridging to work properly.

> **Note**

This behavior is applicable to Cisco Nexus 9300 Switches (Network Forwarding Engine) and Cisco Nexus 9500 Switches with 95xx,96xx,94xx line cards. This behavior is not applicable to Cisco Nexus 9200 Switches, Cisco Nexus 9300-EX and Cisco Nexus 9500 Switches with 9700-EX line cards.

• Port-local VLANs do not support Fabric Extenders (FEX).

• On Cisco Nexus 9364C switches, auto-negotiation may not work on ports 49-64 when bringing up 100G links using QSFP-100G-CR4 cable. To workaround this issue, you must hard-code the speed on ports 49-64 and disable auto-negotiation.

• You may get an error message when you attempt to configure the interface mode to trunk and trunk VLANs simultaneously. On Cisco NX-OS interfaces, the default value of interface mode is access. To implement any trunk related configurations, you must first change the interface mode to trunk and then configure the trunk VLAN ranges.

• On a vPC set up, if the VLAN is a vPC VLAN, the MAC address limit for VLAN and system is not supported.

• All the existing MACs may be flushed and relearnt, when the MAC address table limit is enabled for an interface, VLAN, and/or system.

• MAC address table limit enabled on vPC PO must be consistent across both the peers.
• If you configure MAC address table limit on system, port and VLAN at a time or in any combinations, each one of them will limit the MACs as they are configured. The preference will always be in the following order:
  • Port
  • VLAN
  • System

• MAC address table limit is not supported on vPC peer-links.
• Minimum configurable MAC address table limit is 100 and the maximum configurable limit is 196000.
• When an interface or a VLAN is removed from the set-up, the associated MAC address table limit configuration also gets removed.
• MAC address table limits are not supported on PVLAN interface types.
• When the MAC address table limit exceeds, it floods the traffic, by default.

Default Settings for Layer 2 Interfaces

The following table lists the default settings for device access and trunk port mode parameters.

Table 7: Default Access and Trunk Port Mode Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switchport mode</td>
<td>Access</td>
</tr>
<tr>
<td>Allowed VLANs</td>
<td>1 to 3967, 4048 to 4094</td>
</tr>
<tr>
<td>Access VLAN ID</td>
<td>VLAN1</td>
</tr>
<tr>
<td>Native VLAN ID</td>
<td>VLAN1</td>
</tr>
<tr>
<td>Native VLAN ID tagging</td>
<td>Disabled</td>
</tr>
<tr>
<td>Administrative state</td>
<td>Shut</td>
</tr>
<tr>
<td>SVI autostate</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

Configuring Access and Trunk Interfaces

Note
If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.
Guidelines for Configuring Access and Trunk Interfaces

All VLANs on a trunk must be in the same VDC.

Configuring a VLAN Interface as a Layer 2 Access Port

You can configure a Layer 2 port as an access port. An access port transmits packets on only one, untagged VLAN. You specify which VLAN traffic that the interface carries, which becomes the access VLAN. If you do not specify a VLAN for an access port, that interface carries traffic only on the default VLAN. The default VLAN is VLAN1.

The VLAN must exist before you can specify that VLAN as an access VLAN. The system shuts down an access port that is assigned to an access VLAN that does not exist.

Before you begin

Ensure that you are configuring a Layer 2 interface.

SUMMARY STEPS

1. configure terminal
2. interface ethernet {type slot/port} | {port-channel number}
3. switchport mode [access | trunk]
4. switchport access vlan vlan-id
5. exit
6. show interface
7. no shutdown
8. copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: switch# configure terminal switch(config)#</td>
<td></td>
</tr>
<tr>
<td>Step 2 interface ethernet {type slot/port}</td>
<td>Specifies an interface to configure, and enters interface configuration mode.</td>
</tr>
<tr>
<td>Example: switch(config)# interface ethernet 3/1 switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td>Step 3 switchport mode [access</td>
<td>trunk]</td>
</tr>
<tr>
<td>Example: switch(config-if)# switchport mode access</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Specifies the VLAN for which this access port will carry traffic. If you do not enter this command, the access port carries traffic on VLAN1 only; use this command to change the VLAN for which the access port carries traffic.</td>
</tr>
<tr>
<td><code>switchport access vlan vlan-id</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch(config-if)# switchport access vlan 5</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Exits the interface configuration mode.</td>
</tr>
<tr>
<td><code>exit</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch(config-if)# exit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>(Optional) Displays the interface status and information.</td>
</tr>
<tr>
<td><code>show interface</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch# show interface</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.</td>
</tr>
<tr>
<td><code>no shutdown</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config)# int e3/1</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config-if)# no shutdown</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td><code>copy running-config startup-config</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch(config)# copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>

**Example**

This example shows how to set Ethernet 3/1 as a Layer 2 access port that carries traffic for VLAN 5 only:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# switchport mode access
switch(config-if)# switchport access vlan 5
switch(config-if)#
```

**Configuring Access Host Ports**

- **Note**
  - You should apply the `switchport host` command only to interfaces that are connected to an end station.
  - You can optimize the performance of access ports that are connected to end stations by simultaneously setting that port as an access port. An access host port handles the STP like an edge port and immediately moves to the forwarding state without passing through the blocking and learning states. Configuring an interface as an access host port also disables port channeling on that interface.
See “Configuring Port Channels” section and the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for information about port-channel interfaces.

**Before you begin**

Ensure that you are configuring the correct interface to an interface that is an end station.

**SUMMARY STEPS**

1. `configure terminal`
2. `interface ethernet type slot/port`
3. `switchport host`
4. `exit`
5. `show interface`
6. `no shutdown`
7. `copy running-config startup-config`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
</tbody>
</table>
| Example: | switch# configure terminal  
  switch(config)# |
| **Step 2** | interface ethernet type slot/port |
| Example: | switch(config)# interface ethernet 3/1  
  switch(config-if)# |
| **Step 3** | switchport host |
| Example: | switch(config-if)# switchport host |
| **Step 4** | exit |
| Example: | switch(config-if-range)# exit  
  switch(config)# |
| **Step 5** | show interface |
| Example: | switch# show interface |
### Configuring Layer 2 Interfaces

#### Configuring Trunk Ports

You can configure a Layer 2 port as a trunk port. A trunk port transmits untagged packets for one VLAN plus encapsulated, tagged, packets for multiple VLANs. (See the “IEEE 802.1Q Encapsulation” section for information about encapsulation.)

### Example

This example shows how to set Ethernet 3/1 as a Layer 2 access port with PortFast enabled and port channel disabled:

```bash
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# switchport host
```

### Configuring Trunk Ports

You can configure a Layer 2 port as a trunk port. A trunk port transmits untagged packets for one VLAN plus encapsulated, tagged, packets for multiple VLANs. (See the “IEEE 802.1Q Encapsulation” section for information about encapsulation.)

### Note

The device supports 802.1Q encapsulation only.

### Before you begin

Before you configure a trunk port, ensure that you are configuring a Layer 2 interface.

### SUMMARY STEPS

1. `configure terminal`
2. `interface {type slot/port | port-channel number}`
3. `switchport mode [access | trunk]`
4. `exit`
5. `show interface`
6. `no shutdown`
7. `copy running-config startup-config`

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 6</strong> no shutdown</td>
<td>(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.</td>
</tr>
<tr>
<td><strong>Step 7</strong> copy running-config startup-config</td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>

---

**Example**

This example shows how to set Ethernet 3/1 as a Layer 2 access port with PortFast enabled and port channel disabled:

```bash
switch# configure terminal
switch(config)# int e3/1
switch(config-if)# no shutdown
```
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: switch# configure terminal switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface {type slot/port</td>
<td>port-channel number}</td>
</tr>
<tr>
<td>Example: switch(config)# interface ethernet 3/1 switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> switchport mode [access</td>
<td>trunk]</td>
</tr>
<tr>
<td>Example: switch(config-if)# switchport mode trunk</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> exit</td>
<td>Exits the interface mode.</td>
</tr>
<tr>
<td>Example: switch(config-if)# exit switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> show interface</td>
<td>(Optional) Displays the interface status and information.</td>
</tr>
<tr>
<td>Example: switch# show interface</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> no shutdown</td>
<td>(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.</td>
</tr>
<tr>
<td>Example: switch# configure terminal switch(config)# int e3/1 switch(config-if)# no shutdown</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> copy running-config startup-config</td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td>Example: switch(config)# copy running-config startup-config</td>
<td></td>
</tr>
</tbody>
</table>

### Example

This example shows how to set Ethernet 3/1 as a Layer 2 trunk port:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# switchport mode trunk
switch(config-if)#
```
Configuring the Native VLAN for 802.1Q Trunking Ports

You can configure the native VLAN for 802.1Q trunk ports. If you do not configure this parameter, the trunk port uses the default VLAN as the native VLAN ID.

---

**Note**

You cannot configure an FCoE VLAN as a native VLAN for an Ethernet interface.

---

**SUMMARY STEPS**

1. configure terminal
2. interface {{type slot/port} | {port-channel number}}
3. switchport trunk native vlan vlan-id
4. exit
5. show vlan
6. no shutdown
7. copy running-config startup-config

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
| **Example:** |_switch# configure terminal
| | switch(config)# |
| **Step 2** | interface {{type slot/port} | {port-channel number}} | Specifies an interface to configure, and enters interface configuration mode. |
| **Example:** | switch(config)# interface ethernet 3/1
| | switch(config-if)# |
| **Step 3** | switchport trunk native vlan vlan-id | Sets the native VLAN for the 802.1Q trunk. Valid values are from 1 to 4094, except those VLANs reserved for internal use. The default value is VLAN1. |
| **Example:** | switch(config-if)# switchport trunk native vlan 5 |
| **Step 4** | exit | Exits interface configuration mode. |
| **Example:** | switch(config-if-range)# exit
| | switch(config)# |
| **Step 5** | show vlan | (Optional) Displays the status and information of VLANs. |
| **Example:** | switch# show vlan |
| **Step 6** | no shutdown | (Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the |
Configuring Layer 2 Interfaces

Configuring the Allowed VLANs for Trunking Ports

You can specify the IDs for the VLANs that are allowed on the specific trunk port.

Example

This example shows how to set the native VLAN for the Ethernet 3/1, Layer 2 trunk port to VLAN 5:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# switchport trunk native vlan 5
```

Configuring the Allowed VLANs for Trunking Ports

You can specify the IDs for the VLANs that are allowed on the specific trunk port.

Note

The `switchport trunk allowed vlan vlan-list` command replaces the current VLAN list on the specified port with the new list. You are prompted for confirmation before the new list is applied.

If you are doing a copy and paste of a large configuration, you might see some failures because the CLI is waiting for a confirmation before accepting other commands. To avoid this problem, you can disable prompting by using the `terminal dont-ask` command before you paste the configuration.

Before you begin

Before you configure the allowed VLANs for the specified trunk ports, ensure that you are configuring the correct interfaces and that the interfaces are trunks.

Note

You can change the block of VLANs reserved for internal use. See the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for more information about changing the reserved VLANs.

SUMMARY STEPS

1. configure terminal
2. interface {ethernet slot/port | port-channel number}
3. switchport trunk allowed vlan {vlan-list add vlan-list | all | except vlan-list | none | remove vlan-list}
4. exit
5. show vlan
### Configuring Layer 2 Interfaces

#### Configuring the Allowed VLANs for Trunking Ports

6. no shutdown  
7. copy running-config startup-config

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
|      | Example: switch# configure terminal  
|      | switch(config)# | |
| 2    | interface {ethernet slot/port | port-channel number} | Specifies an interface to configure, and enters interface configuration mode. |
|      | Example: switch(config)# interface ethernet 3/1 | |
| 3    | switchport trunk allowed vlan {vlan-list add vlan-list | all | except vlan-list | none | remove vlan-list} | Sets the allowed VLANs for the trunk interface. The default is to allow all VLANs on the trunk interface: 1 to 3967 and 4048 to 4094. VLANs 3968 to 4047 are the default VLANs reserved for internal use by default. By default, all VLANs are allowed on all trunk interfaces. The default reserved VLANs are 3968 to 4094, and you can change the block of reserved VLANs. See the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for more information. |
|      | Example: switch(config-if)# switchport trunk allowed vlan add 15-20# | |
| 4    | exit | Exits the interface mode. |
|      | Example: switch(config-if)# exit  
|      | switch(config)# | |
| 5    | show vlan | (Optional) Displays the status and information for VLANs. |
|      | Example: switch# show vlan | |
| 6    | no shutdown | (Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state. |
|      | Example: switch# configure terminal  
|      | switch(config)# int e3/1  
|      | switch(config-if)# no shutdown | |
| 7    | copy running-config startup-config | (Optional) Copies the running configuration to the startup configuration. |
|      | Example: | |
Example

This example shows how to add VLANs 15 to 20 to the list of allowed VLANs on the Ethernet 3/1, Layer 2 trunk port:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# switchport trunk allowed vlan 15-20
switch(config-if)#
```

Configuring MAC addresses Limitation on a VLAN

Cisco Nexus 9500 Series switches with N9K-X9636C-R, N9K-X9636Q-R, N9K-X9636C-RX line cards provides the ability to set an upper limit for the number of MAC addresses that can reside inside MAC address table of a Line-card Expansion-module (LEM). You can configure the limitations at System, VLAN, port, trunk and tunnel levels. For instance if the specified VLAN limitation is 2000 MACs, the Layer 2 Forwarding Manager (L2FM) accepts the first 2000 MACs it receives and reject the remaining MACs. To configure MAC address limitation on VLAN, complete the following steps:

**Note**

Though this method stops learning MAC address learning via VLAN or system after the limit is reached, the traffic passes through the interface.

**SUMMARY STEPS**

1. switch# configure terminal
2. switch(config)# mac address-table limit system value
3. switch(config)# mac address-table limit vlan value
4. switch(config)# exit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# mac address-table limit system value</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config)# mac address-table limit vlan value</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>switch(config)# exit</td>
</tr>
</tbody>
</table>
Example

This example shows how to configure the upper limit for MAC learning at system and VLAN levels:

```bash
switch# configure terminal
switch(config)# mac address-table limit system 10000
Configuring Mac address limit will result in flushing existing Macs in the specified VLAN/System. Proceed(yes/no)? [no] yes
switch(config)# mac address-table limit vlan 30 3000
Configuring Mac address limit will result in flushing existing Macs in the specified VLAN/System. Proceed(yes/no)? [no] yes
switch(config)# exit
```

This example shows how to display the MAC address limitations:

```bash
switch# configure terminal
switch(config)# sh mac address-table limit
System Limit: 10000

Vlan  Learning Limit
-----  ----------
 1      196000
 20     196000
 30     3000
 100    196000
switch(config)# exit
```

Configuring MAC Addresses Limitation on a Port

Beginning Cisco NX-OS Release 9.2(3), Cisco Nexus 9500 Series switches with N9K-X9636C-RX, N3K-C3636C-R and N3K-C36180YC-R line cards provides the ability to set an upper limit for the number of MAC addresses that can be learnt by each port. For example, if the specified VLAN limitation is 2000 MACs, the Layer 2 Forwarding Manager (L2FM) accepts the first 2000 MACs it receives and reject the remaining MACs. To configure MAC address limitation on an interface, follow these steps:

**SUMMARY STEPS**

1. switch# configure terminal
2. switch(config)# mac address-table limit interface port-channel value
3. switch(config)# show mac address-table limit interf
4. switch(config)# exit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>switch(config)# mac address-table limit interface port-channel value</td>
<td>Specifies an upper limit for MAC learning at port level.</td>
</tr>
</tbody>
</table>
## Configuring switchport isolated

The switchport isolated feature is supported.

### Note

The switchport isolated feature does not support FEX interfaces nor port-channel members.

---

### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config)# show mac address-table limit interf</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>switch(config)# exit</td>
</tr>
</tbody>
</table>

### Example

This example shows how to configure the upper limit for MAC learning at port levels:

```markdown
switch# configure terminal
switch(config)# mac address-table limit interface port-channel 2 1000
Configuring Mac address limit will result in flushing existing Macs in the specified VLAN/System. Proceed(yes/no)? [no] yes
switch(config)# exit
```

This example shows how to display the MAC address limitations:

```markdown
switch# configure terminal
switch(config)# show mac address-table limit interf
Interface  Conf Limit  Curr Count  Cfg Action  Currently
-----------  ---------  ---------  ----------  -------
Vlan1        196000    0         Flood     Flooding Unknown SA
Vlan341      196000    0         Flood     Flooding Unknown SA
Vlan342      196000    0         Flood     Flooding Unknown SA
Vlan343      196000    0         Flood     Flooding Unknown SA
Vlan344      196000    0         Flood     Flooding Unknown SA
Vlan345      196000    0         Flood     Flooding Unknown SA
Vlan346      196000    0         Flood     Flooding Unknown SA
Vlan347      196000    0         Flood     Flooding Unknown SA
Vlan348      196000    0         Flood     Flooding Unknown SA
Vlan349      196000    0         Flood     Flooding Unknown SA
Vlan350      196000    0         Flood     Flooding Unknown SA
port-channel1 196000    0         Flood     Flooding Unknown SA
port-channel2  1000     0         Flood     Flooding Unknown SA
port-channel11 196000    0         Flood     Flooding Unknown SA
port-channel12 196000    0         Flood     Flooding Unknown SA
port-channel13 196000    0         Flood     Flooding Unknown SA
port-channel1601 196000    0         Flood     Flooding Unknown SA
port-channel1603 196000    0         Flood     Flooding Unknown SA
port-channel1888 196000    0         Flood     Flooding Unknown SA
Ethernet1/6   196000    0         Flood     Flooding Unknown SA
Ethernet1/15   196000    0         Flood     Flooding Unknown SA
Ethernet1/35   196000    0         Flood     Flooding Unknown SA
BF2(config)# exit
```
Physical interfaces with different switchport isolated configuration are not allowed in a port-channel.

### SUMMARY STEPS

1. configure terminal
2. interface {{ethernet slot/port} | {port-channel number}}
3. switchport isolated
4. show running-config interface port-channel port-channel-number

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: switch# configure terminal switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface {{ethernet slot/port}</td>
<td>{port-channel number}}</td>
</tr>
<tr>
<td>Example: switch(config)# interface ethernet 3/1 switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> switchport isolated</td>
<td>Enables the switchport isolated feature.</td>
</tr>
<tr>
<td>Example: switch(config-if)# switchport isolated</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> show running-config interface port-channel port-channel-number</td>
<td>(Optional) Displays the interface status and information.</td>
</tr>
</tbody>
</table>

### Configuring a Default Interface

The default interface feature allows you to clear the existing configuration of multiple interfaces such as Ethernet, loopback, VLAN network, port-channel, and tunnel interfaces. All user configuration under a specified interface will be deleted. You can optionally create a checkpoint before clearing the interface configuration so that you can later restore the deleted configuration.

**Note**

The default interface feature is not supported for management interfaces because the device could go to an unreachable state.

If the speed group is configured, the **default interface** command displays the following error:

ERROR: default interface is not supported as speed-group is configured
### SUMMARY STEPS

1. configure terminal
2. default interface `int-if[checkpoint name]`
3. exit
4. show interface
5. no shutdown

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
| **Example:**
  - `switch# configure terminal`
  - `switch(config)#` |
| **Step 2** default interface `int-if[checkpoint name]` | Deletes the configuration of the interface and restores the default configuration. Use the `?` keyword to display the supported interfaces. Use the `checkpoint` keyword to store a copy of the running configuration of the interface before clearing the configuration. |
| **Example:**
  - `switch(config)# default interface ethernet 3/1 checkpoint test8` |
| **Step 3** exit | Exits global configuration mode. |
| **Example:**
  - `switch(config)# exit`
  - `switch(config)#` |
| **Step 4** show interface | (Optional) Displays the interface status and information. |
| **Example:**
  - `switch# show interface` |
| **Step 5** no shutdown | (Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state. |
| **Example:**
  - `switch# configure terminal`
  - `switch(config)# int e3/1`
  - `switch(config-if)# no shutdown` |

#### Example

This example shows how to delete the configuration of an Ethernet interface while saving a checkpoint of the running configuration for rollback purposes:

```
switch# configure terminal
switch(config)# default interface ethernet 3/1 checkpoint test8
.......Done
switch(config)#
```
Configuring SVI Autostate Exclude

You can configure the SVI Autostate Exclude feature on an Ethernet interface or a port channel. You can use the Autostate Exclude option to enable or disable the port from bringing up or down the SVI calculation and applying it to all VLANs that are enabled on the selected port. You can also use the SVI Autostate Exclude VLAN feature to exclude a VLAN from the autostate excluded interface.

SUMMARY STEPS

1. configure terminal
2. interface {{type slot/port} | {port-channel number}}
3. switchport
4. [no] switchport autostate exclude
5. [no] switchport autostate exclude vlan {vlan id | all | except}
6. exit
7. show running-config interface {{type slot/port} | {port-channel number}}
8. no shutdown
9. copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
</tbody>
</table>
| Example: | switch# configure terminal  
switch(config)# |
| **Step 2** | interface {{type slot/port} | {port-channel number}} |
| Example: | switch(config)# interface ethernet 3/1  
switch(config-if)# |
| **Step 3** | switchport |
| Example: | switch(config-if)# switchport |
| **Step 4** | [no] switchport autostate exclude |
| Example: | switch(config-if)# switchport autostate exclude |
| **Step 5** | [no] switchport autostate exclude vlan {vlan id | all | except} |
| Example: | switch(config-if)# switchport autostate exclude vlan 10 |
### Command or Action

**Step 6**

- **exit**
  - Example:
    ```
    switch(config-if)# exit
    switch(config)#
    ```

- **Purpose**: Exits the interface configuration mode.

**Step 7**

- **show running-config interface** `{type slot/port} | {port-channel number}`
  - Example:
    ```
    switch(config)# show running-config interface ethernet 3/1
    ```

- **Purpose**: (Optional) Displays configuration information about the specified interface.

**Step 8**

- **no shutdown**
  - Example:
    ```
    switch# configure terminal
    switch(config)# int e3/1
    switch(config-if)# no shutdown
    ```

- **Purpose**: (Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

**Step 9**

- **copy running-config startup-config**
  - Example:
    ```
    switch(config)# copy running-config startup-config
    ```

- **Purpose**: (Optional) Copies the running configuration to the startup configuration.

### Example

This example shows how to exclude a port from the VLAN interface link-up calculation on the Cisco NX-OS device:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# switchport
switch(config-if)# switchport autostate exclude
```

This example shows how to exclude a VLAN from the auto-excluded interface:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# switchport
switch(config-if)# switchport autostate exclude
switch(config-if)# switchport autostate exclude vlan 10
```

### Configuring SVI Autostate Disable for the System

You can manage an SVI with the SVI autostate feature. You can configure the SVI autostate disable feature to keep an SVI up even if no interface is up in the corresponding VLAN. (Similarly, configure the SVI autostate enable feature so an SVI goes down when no interface is up in the corresponding VLAN). Use this procedure to configure this feature for the entire system.
The `system default interface-vlan autostate` command enables the SVI autostate feature.

### SUMMARY STEPS

1. `configure terminal`
2. `[no] system default interface-vlan autostate`
3. `no shutdown`
4. `show running-config [all]`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**        | **configure terminal**
| **Example:**      | `switch# configure terminal`
|                   | `switch(config)#` |
| **Step 2**        | `[no] system default interface-vlan autostate`
| **Example:**      | `switch(config)# no system default interface-vlan autostate` |
| **Step 3**        | `no shutdown`
| **Example:**      | `switch# configure terminal`
|                   | `switch(config)# int e3/1`
|                   | `switch(config-if)# no shutdown` |
| **Step 4**        | `show running-config [all]`
| **Example:**      | `switch(config)# show running-config` |

**Example**

This example shows how to disable the default autostate behavior on the Cisco NX-OS device:

```
switch# configure terminal
switch(config)# no system default interface-vlan autostate
switch(config)# show running-config
```

### Configuring SVI Autostate Disable Per SVI

You can configure SVI autostate enable or disable on individual SVIs. The SVI-level setting overrides the system-level SVI autostate configuration for that particular SVI.
### SUMMARY STEPS

1. `configure terminal`  
2. `feature interface-vlan`  
3. `interface vlan vlan-id`  
4. `[no] autostate`  
5. `exit`  
6. `show running-config interface vlan vlan-id`  
7. `no shutdown`  
8. `show startup-config interface vlan vlan-id`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
  **Example:**  
  `switch# configure terminal`  
  `switch(config)#`  |
| **Step 2** | Enables VLAN interface mode. |
|  
  **Example:**  
  `switch(config)# feature interface-vlan`  |
| **Step 3** | Creates a VLAN interface and enters interface configuration mode. The range is from 1 and 4094. |
|  
  **Example:**  
  `switch(config-if)# interface vlan10`  
  `switch(config)#`  |
| **Step 4** | By default, enables the SVI autostate feature on specified interface.  
By disabling, the default settings, use the no form of this command. |
|  
  **Example:**  
  `switch(config-if)# no autostate`  |
| **Step 5** | Exits the interface configuration mode. |
|  
  **Example:**  
  `switch(config-if)# exit`  
  `switch(config)#`  |
| **Step 6** | (Optional) Displays the running configuration for the specified VLAN interface. |
|  
  **Example:**  
  `switch(config)# show running-config interface vlan10`  |
| **Step 7** | (Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the |
|  
  **Example:**  
  `switch(config)# show running-config interface vlan10`  |
### Configuring Layer 2 Interfaces

#### Configuring the Device to Tag Native VLAN Traffic

When you are working with 802.1Q trunked interfaces, you can maintain the tagging for all packets that enter with a tag that matches the value of the native VLAN ID and drops all untagged traffic (you will still carry control traffic on that interface). This feature applies to the entire device; you cannot apply it to selected VLANs on a device.

The `vlan dot1q tag native` command changes the behavior of all native VLAN ID interfaces on all trunks on the device.

---

**Note**

If you enable 802.1Q tagging on one device and disable it on another device, all traffic is dropped on the device and this feature is disabled. You must configure this feature identically on each device.

---

**SUMMARY STEPS**

1. `configure terminal`
2. `vlan dot1q tag native`
3. `exit`
4. `show vlan`
5. `no shutdown`
6. `copy running-config startup-config`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>switch(config)# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config)# int e3/1</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config-if)# no shutdown</code></td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

```
switch# configure terminal
switch(config)# int e3/1
switch(config-if)# no shutdown
```

Port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

**Step 8**

```
show startup-config interface vlan vlan-id
```

(Optional) Displays the VLAN configuration in the startup configuration.

**Example:**

```
switch(config)# show startup-config interface vlan10
```

This example shows how to disable the default autostate behavior on an individual SVI:

```
switch# configure terminal
switch(config)# feature interface-vlan
switch(config)# interface vlan10
switch(config-if)# no autostate
```

---

Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide, Release 9.2(x)
### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>switch# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config)#</code></td>
<td></td>
</tr>
</tbody>
</table>

**Step 2**  
`vlan dot1q tag native`  
**Example:**  
```
switch(config)# vlan dot1q tag native
```

Modifies the behavior of a 802.1Q trunked native VLAN ID interface. The interface maintains the taggings for all packets that enter with a tag that matches the value of the native VLAN ID and drops all untagged traffic. The control traffic is still carried on the native VLAN. The default is disabled.

**Step 3**  
`exit`  
**Example:**  
```
switch(config-if-range)# exit
switch(config)#
```

Exits the interface configuration mode.

**Step 4**  
`show vlan`  
**Example:**  
```
switch# show vlan
```

(Optional) Displays the status and information for VLANs.

**Step 5**  
`no shutdown`  
**Example:**  
```
switch# configure terminal
switch(config)# int e3/1
switch(config-if)# no shutdown
```

(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

**Step 6**  
`copy running-config startup-config`  
**Example:**  
```
switch(config)# copy running-config startup-config
```

(Optional) Copies the running configuration to the startup configuration.

### Example

This example shows how to change the behavior of the native VLAN on an 802.1Q trunked interface to maintain the tagged packets and drop all untagged traffic (except control traffic):

```
switch# configure terminal
switch(config)# vlan dot1q tag native
switch#
```

### Configuring Interface Breakout Profile for 50-G Interfaces in a 16-Slot Chassis

The interface breakout profile is needed to breakout high bandwidth 100-G ports into two 50-G interfaces for slot 8 to 16 in the Cisco Nexus 9516 switch for -EX line cards.

### SUMMARY STEPS

1. configure terminal  
2. (Optional) interface breakout-profile 50g-2x-only
3. copy running-config startup-config
4. reload
5. interface breakout module module-number port port-range map [10g-4x | 25g-4x | 50g-2x]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>(Optional) interface breakout-profile 50g-2x-only</td>
<td>This command is required to breakout slots 8 to 16. It is not required for slots 1 to 7.</td>
</tr>
<tr>
<td></td>
<td>Example: switch(config)# interface breakout-profile 50g-2x-only Warning: Please save config and reload the switch for breakout-profile config to take effect Please save config and reload the switch for the configuration to take effect</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>copy running-config startup-config</td>
<td>Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td></td>
<td>Example: switch(config-inf)# copy running-config startup-config [########################################] 100% Copy complete, now saving to disk (please wait)... Copy complete.</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>reload</td>
<td>Reboots the switch.</td>
</tr>
<tr>
<td></td>
<td>Example: switch(config-inf)# reload This command will reboot the system. (y/n)? [n] y</td>
<td>Note: After the switch reloads and the modules are up, enter the following CLI for any module or ports to breakout.</td>
</tr>
<tr>
<td>Step 5</td>
<td>interface breakout module module-number port port-range map [10g-4x</td>
<td>25g-4x</td>
</tr>
<tr>
<td></td>
<td>Example: switch(config)# interface breakout module 1 port 1-32 map 50g-2x</td>
<td></td>
</tr>
</tbody>
</table>

Changing the System Default Port Mode to Layer 2

You can set the system default port mode to Layer 2 access ports.

SUMMARY STEPS

1. configure terminal
2. system default switchport [shutdown]
3. exit
### Configuring Layer 2 Interfaces

#### Changing the System Default Port Mode to Layer 2

1. **Configure Terminal**
   - `configure terminal`
   - **Example:**
     ```
     switch# configure terminal
     switch(config)#
     ```

2. **System Default Switchport [shutdown]**
   - `system default switchport [shutdown]`
   - **Example:**
     ```
     switch(config-if)# system default switchport
     ```
   - **Note:** When the `system default switchport shutdown` command is issued:
     - Any FEX HIFs that are not configured with `no shutdown` are shutdown. To avoid the shutdown, configure the FEX HIFs with `no shut`.
     - Any Layer 2 port that is not specifically configured with `no shutdown` are shutdown. To avoid the shutdown, configure the Layer 2 port with `no shut`.

3. **Exit**
   - `exit`
   - **Example:**
     ```
     switch(config-if)# exit
     switch(config)#
     ```

4. **Show Interface Brief**
   - `show interface brief`
   - **Example:**
     ```
     switch# show interface brief
     ```

5. **No Shutdown**
   - `no shutdown`
   - **Example:**
     ```
     switch# configure terminal
     switch(config)# int e3/1
     switch(config-if)# no shutdown
     ```

6. **Copy Running-Config Startup-Config**
   - `copy running-config startup-config`
   - **Example:**
     ```
     switch(config)# copy running-config startup-config
     ```

---

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
| `configure terminal` | `switch# configure terminal
switch(config)#` |
| **Step 2**        | system default switchport [shutdown] |
| **Example:**      | Sets the default port mode for all interfaces on the system to Layer 2 access port mode and enters interface configuration mode. By default, all the interfaces are Layer 3. |
| `system default switchport` | `switch(config-if)# system default switchport` |
| **Step 3**        | exit  |
| **Example:**      | Exits the interface configuration mode. |
| `exit` | `switch(config-if)# exit
switch(config)#` |
| **Step 4**        | show interface brief |
| **Example:**      | (Optional) Displays the status and information for interfaces. |
| `show interface brief` | `switch# show interface brief` |
| **Step 5**        | no shutdown |
| **Example:**      | (Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state. |
| `no shutdown` | `switch# configure terminal
switch(config)# int e3/1
switch(config-if)# no shutdown` |
| **Step 6**        | copy running-config startup-config |
| **Example:**      | (Optional) Copies the running configuration to the startup configuration. |
| `copy running-config startup-config` | `switch(config)# copy running-config startup-config` |
Example
This example shows how to set the system ports to be Layer 2 access ports by default:

```
switch# configure terminal
switch(config-if)# system default switchport
switch(config-if)#
```

Verifying the Interface Configuration

To display access and trunk interface configuration information, perform one of the following tasks.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>`show interface ethernet slot/port [brief</td>
<td>counters</td>
</tr>
<tr>
<td><code>show interface brief</code></td>
<td>Displays interface configuration information, including the mode.</td>
</tr>
<tr>
<td><code>show interface switchport</code></td>
<td>Displays information, including access and trunk interface, information for all Layer 2 interfaces.</td>
</tr>
<tr>
<td>`show interface trunk [module module-number</td>
<td>vlan vlan-id]`</td>
</tr>
<tr>
<td><code>show interface capabilities</code></td>
<td>Displays information about the capabilities of the interfaces.</td>
</tr>
<tr>
<td><code>show running-config [all]</code></td>
<td>Displays information about the current configuration. The <code>all</code> command displays the default and current configurations.</td>
</tr>
<tr>
<td><code>show running-config interface ethernet slot/port</code></td>
<td>Displays configuration information about the specified interface.</td>
</tr>
<tr>
<td><code>show running-config interface port-channel slot/port</code></td>
<td>Displays configuration information about the specified port-channel interface.</td>
</tr>
<tr>
<td><code>show running-config interface vlan vlan-id</code></td>
<td>Displays configuration information about the specified VLAN interface.</td>
</tr>
</tbody>
</table>

Monitoring the Layer 2 Interfaces

Use the following commands to display Layer 2 interfaces:
### Configuration Examples for Access and Trunk Ports

This example shows how to configure a Layer 2 access interface and assign the access VLAN mode for that interface:

```bash
switch# configure terminal
switch(config)# interface ethernet 2/30
switch(config-if)# switchport
switch(config-if)# switchport mode access
switch(config-if)# switchport access vlan 5
switch(config-if)#
```

This example shows how to configure a Layer 2 trunk interface, assign the native VLAN and the allowed VLANs, and configure the device to tag the native VLAN traffic on the trunk interface:

```bash
switch# configure terminal
switch(config)# interface ethernet 2/35
switch(config-if)# switchport
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk native vlan 10
switch(config-if)# switchport trunk allowed vlan 5, 10
switch(config-if)# exit
switch(config)# vlan dot1q tag native
switch(config)#
```

### Related Documents

<table>
<thead>
<tr>
<th>Related Documents</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring Layer 3 interfaces</td>
<td>Configuring Layer 2 Interfaces section</td>
</tr>
<tr>
<td>Port Channels</td>
<td>Configuring Port Channels section</td>
</tr>
<tr>
<td>VLANs, private VLANs, and STP</td>
<td>Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide</td>
</tr>
<tr>
<td>System management</td>
<td>Cisco Nexus 9000 Series NX-OS System Management Configuration Guide</td>
</tr>
</tbody>
</table>
### Related Documents

<table>
<thead>
<tr>
<th>Related Documents</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>High availability</td>
<td><em>Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide</em></td>
</tr>
<tr>
<td>Licensing</td>
<td><em>Cisco NX-OS Licensing Guide</em></td>
</tr>
<tr>
<td>Release Notes</td>
<td><em>Cisco Nexus 9000 Series NX-OS Release Notes</em></td>
</tr>
</tbody>
</table>
Configuring Layer 3 Interfaces

- About Layer 3 Interfaces, on page 91
- Licensing Requirements for Layer 3 Interfaces, on page 96
- Prerequisites for Layer 3 Interfaces, on page 97
- Guidelines and Limitations for Layer 3 Interfaces, on page 97
- Default Settings, on page 98
- Configuring Layer 3 Interfaces, on page 98
- Verifying the Layer 3 Interfaces Configuration, on page 119
- Monitoring the Layer 3 Interfaces, on page 120
- Configuration Examples for Layer 3 Interfaces, on page 121
- Related Documents, on page 123

About Layer 3 Interfaces

Layer 3 interfaces forward IPv4 and IPv6 packets to another device using static or dynamic routing protocols. You can use Layer 3 interfaces for IP routing and inter-VLAN routing of Layer 2 traffic.

Routed Interfaces

You can configure a port as a Layer 2 interface or a Layer 3 interface. A routed interface is a physical port that can route IP traffic to another device. A routed interface is a Layer 3 interface only and does not support Layer 2 protocols, such as the Spanning Tree Protocol (STP).

All Ethernet ports are routed interfaces by default. You can change this default behavior with the CLI setup script.

Note

The default behavior varies based on the type of switch (Cisco Nexus 9300, Cisco Nexus 9500, or Cisco Nexus 3164).

Note

Cisco Nexus 9300 Series switches (except Cisco Nexus 9332 switch) have a Layer 2 default mode.
You can assign an IP address to the port, enable routing, and assign routing protocol characteristics to this routed interface.

You can also create a Layer 3 port channel from routed interfaces. For more information about port channels, see the “Configuring Port Channels” section.

Routed interfaces and subinterfaces support exponentially decayed rate counters. Cisco NX-OS tracks the following statistics with these averaging counters:

- Input packets/sec
- Output packets/sec
- Input bytes/sec
- Output bytes/sec

**Subinterfaces**

You can create virtual subinterfaces on a parent interface configured as a Layer 3 interface. A parent interface can be a physical port.

Subinterfaces divide the parent interface into two or more virtual interfaces on which you can assign unique Layer 3 parameters such as IP addresses and dynamic routing protocols. The IP address for each subinterface should be in a different subnet from any other subinterface on the parent interface.

You create a subinterface with a name that consists of the parent interface name (for example, Ethernet 2/1) followed by a period and then by a number that is unique for that subinterface. For example, you could create a subinterface for Ethernet interface 2/1 named Ethernet 2/1.1 where .1 indicates the subinterface.

Cisco NX-OS enables subinterfaces when the parent interface is enabled. You can shut down a subinterface independent of shutting down the parent interface. If you shut down the parent interface, Cisco NX-OS shuts down all associated subinterfaces as well.

One use of subinterfaces is to provide unique Layer 3 interfaces to each virtual local area network (VLAN) supported by the parent interface. In this scenario, the parent interface connects to a Layer 2 trunking port on another device. You configure a subinterface and associate the subinterface to a VLAN ID using 802.1Q trunking.

The following figure shows a trunking port from a switch that connects to router B on interface E 2/1. This interface contains three subinterfaces that are associated with each of the three VLANs carried by the trunking port.

*Figure 4: Subinterfaces for VLANs*

For more information about VLANs, see the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide.
Limitations for Subinterfaces

The following are the limitations for subinterfaces:

• Statistics for subinterfaces are not supported.

VLAN Interfaces

A VLAN interface, or switch virtual interface (SVI), is a virtual routed interface that connects a VLAN on the device to the Layer 3 router engine on the same device. Only one VLAN interface can be associated with a VLAN, but you need to configure a VLAN interface for a VLAN only when you want to route between VLANs or to provide IP host connectivity to the device through a virtual routing and forwarding (VRF) instance that is not the management VRF. When you enable VLAN interface creation, Cisco NX-OS creates a VLAN interface for the default VLAN (VLAN 1) to permit remote switch administration.

You must enable the VLAN network interface feature before you can see configure it. The system automatically takes a checkpoint prior to disabling the feature, and you can roll back to this checkpoint. See the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide for information on rollbacks and checkpoints.

You cannot delete the VLAN interface for VLAN 1.

You can route across VLAN interfaces to provide Layer 3 inter-VLAN routing by configuring a VLAN interface for each VLAN that you want to route traffic to and assigning an IP address on the VLAN interface. For more information about IP addresses and IP routing, see the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.

The following figure shows two hosts connected to two VLANs on a device. You can configure VLAN interfaces for each VLAN that allows Host 1 to communicate with Host 2 using IP routing between the VLANs. VLAN 1 communicates at Layer 3 over VLAN interface 1 and VLAN 10 communicates at Layer 3 over VLAN interface 10.

Figure 5: Connecting Two VLANs with VLAN interfaces

Changing VRF Membership for an Interface

When you enter the `vrf member` command under an interface, you receive an alert regarding the deletion of interface configurations and to notify the clients/listeners (such as CLI-Server) to delete configurations with respect to the interface.
Entering the **system vrf-member-change retain-l3-config** command enables the retention of the Layer 3 configuration when the VRF member changes on the interface. It does this by sending notification to the clients/listeners to store (buffer) the existing configurations, delete the configurations from the old vrf context, and reapply the stored configurations under the new VRF context.

**Note**

When the **system vrf-member-change retain-l3-config** command is enabled, the Layer 3 configuration is not deleted and remains stored (buffered). When this command is not enabled (default mode), the Layer 3 configuration is not retained when the VRF member changes.

You can disable the retention of the Layer 3 configuration with the **no system vrf-member-change retain-l3-config** command. In this mode, the Layer 3 configuration is not retained when the VRF member changes.

**Notes About Changing VRF Membership for an Interface**

- Momentary traffic loss may occur when changing the VRF name.
- Only the configurations under the interface level are processed when the **system vrf-member-change retain-l3-config** command is enabled. You must manually process any configurations at the router level to accommodate routing protocols after a VRF change.
- The **system vrf-member-change retain-l3-config** command supports interface level configurations with:
  - Layer 3 configurations maintained by the CLI Server, such as **ip address** and **ipv6 address** (secondary) and all OSPF/ISIS/EIGRP CLIs available under the interface configuration.
  - HSRP
  - DHCP Relay Agent CLIs, such as **ip dhcp relay address** [use-vrf] and **ipv6 dhcp relay address** [use-vrf].
- For DHCP:
  - As a best practice, the client and server interface VRF should be changed one at a time. Otherwise, the DHCP packets cannot be exchanged on the relay agent.
  - When the client and server are in different VRFs, use the **ip dhcp relay address** [use-vrf] command to exchange the DHCP packets in the relay agent over the different VRFs.

**Loopback Interfaces**

A loopback interface is a virtual interface with a single endpoint that is always up. Any packet transmitted over a loopback interface is immediately received by this interface. Loopback interfaces emulate a physical interface. You can configure up to 1024 loopback interfaces, numbered 0 to 1023.

You can use loopback interfaces for performance analysis, testing, and local communications. Loopback interfaces can act as a termination address for routing protocol sessions. This loopback configuration allows routing protocol sessions to stay up even if some of the outbound interfaces are down.
IP Unnumbered

The IP unnumbered feature enables the processing of IP packets on a point to point (p2p) interface without explicitly configuring a unique IP address on it. This approach borrows an IP address from another interface and conserves address space on point to point links.

Any interface which conforms to the point to point mode can be used as an IP unnumbered interface. The IP unnumbered feature is supported only on Ethernet interfaces and sub-interfaces. The borrowed interface can only be a loopback interface and is known as the numbered interface.

A loopback interface is ideal as a numbered interface in that it is always functionally up. However, because loopback interfaces are local to a switch/router, the reachability of unnumbered interfaces first needs to be established through static routes or by using an interior gateway protocol, such as OSPF or ISIS.

Configuring IP unnumbered interfaces for port channels is supported on all Cisco Nexus 9000 Series switches.

MAC-Embedded IPv6 Address

BGP allows an IPv4 prefix to be carried over an IPv6 next hop. The IPv6 next hop is leveraged to remove neighbor discovery (ND)-related traffic from the network. To do this, the MAC address is embedded in the IPv6 address. Such an address is called a MAC-embedded IPv6 (MEv6) address. The router extracts the MAC address directly from the MEv6 address instead of going through ND. Local interface and next-hop MAC addresses are extracted from the IPv6 addresses.

On MEv6-enabled IPv6 interfaces, the same MEv6-extracted MAC address is used for IPv4 traffic as well. MEv6 is supported on all Layer 3-capable interfaces except switch virtual interfaces (SVIs).

Important

When MEv6 is enabled on an interface, ping6 to the IPv6 link local address, OSPFv3, and BFDv6 are not supported on that interface.

High Availability

Layer 3 interfaces support stateful and stateless restarts. After the switchover, Cisco NX-OS applies the runtime configuration after the switchover.

See the Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide for complete information about high availability.

Virtualization Support

Layer 3 interfaces support Virtual Routing and Forwarding instances (VRFs). VRFs exist within virtual device contexts (VDCs). By default, Cisco NX-OS places you in the default VDC and default VRF.

Note

You must assign an interface to a VRF before you configure the IP address for that interface.
DHCP Client

Cisco NX-OS supports DHCP client for IPv4 and IPv6 addresses on SVIs, physical Ethernet, and management interfaces. You can configure the IP address of a DHCP client by using the `ip address dhcp` or `ipv6 address dhcp` command. These commands send a request from the DHCP client to the DHCP server soliciting an IPv4 or IPv6 address from the DHCP server. The DHCP client on the Cisco Nexus switch identifies itself to the DHCP server. The DHCP server uses this identifier to send the IP address back to the DHCP client.

When a DHCP client is configured on the SVI with the DHCP server sending router and DNS options, the `ip route 0.0.0.0/0 router-ip` and `ip name-server dns-ip` commands are configured on the switch automatically.

Limitations for Using DHCP Client on Interfaces

The following are the limitations for using DHCP client on interfaces:

- This feature is supported only on physical Ethernet interfaces, management interfaces, and SVIs.
- This feature is supported on non-default virtual routing and forwarding (VRF) instances.
- The DNS server and default router option-related configurations are saved in the startup configuration when you enter the `copy running-config startup-config` command. When you reload the switch, if this configuration is not applicable, you might have to remove it.
- You can configure a maximum of six DNS servers on the switch, which is a switch limitation. This maximum number includes the DNS servers configured by the DHCP client and the DNS servers configured manually.
- If the number of DNS servers configured on the switch is more than six, and if you get a DHCP offer for an SVI with DNS option set, the IP address is not assigned to the SVI.
- A Cisco Nexus 9000 Series switch supports a maximum of 10 IPv4 and 10 IPv6 DHCP clients.
- DHCP relay and DHCP client configurations are incompatible and are not supported on the same switch. You must remove the DHCP relay configuration before configuring the DHCP Client on an interface.
- When DHCP snooping is enabled on the VLAN whose SVI is configured with the DHCP client, the DHCP snooping is not enforced on the SVI DHCP client.
- When configuring the IPv6 DHCP client, you must configure with the `ipv6 address use-link-local-only` command before the `ipv6 address dhcp` command.

Licensing Requirements for Layer 3 Interfaces

The following table shows the licensing requirements for this feature:

<table>
<thead>
<tr>
<th>Product</th>
<th>License Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco NX-OS</td>
<td>Layer 3 interfaces require no license. Any feature not included in a license package is bundled with the Cisco NX-OS image and is provided at no extra charge to you.</td>
</tr>
</tbody>
</table>
Prerequisites for Layer 3 Interfaces

Layer 3 interfaces have the following prerequisites:

• You are familiar with IP addressing and basic configuration. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information about IP addressing.

Guidelines and Limitations for Layer 3 Interfaces

Layer 3 interfaces have the following configuration guidelines and limitations:

• show commands with the internal keyword are not supported.

• Configuring a subinterface on a port-channel interface is not supported.

• If you change a Layer 3 interface to a Layer 2 interface, Cisco NX-OS shuts down the interface, reenables the interface, and removes all configuration specific to Layer 3.

• If you change a Layer 2 interface to a Layer 3 interface, Cisco NX-OS shuts down the interface, reenables the interface, and deletes all configuration specific to Layer 2.

• The Dynamic Host Configuration Protocol (DHCP) option is not supported when configuring a subinterface on a port-channel interface.

• When an IP unnumbered interface is configured, a loopback interface should be in the same VRF as the IP unnumbered interface.

• An admin-shutdown command on a loopback interface that is a numbered interface does not bring down the IP unnumbered interface. This means that the routing protocols running over the IP unnumbered interface continue to be up.

• Static routes running over the IP unnumbered interface should use pinned static routes.

Note

The IP unnumbered interface through which the route is resolved needs to be specified.

• An IP unnumbered interface is supported only on physical and sub-interfaces.

• Only loopback interfaces can use unnumbered interfaces as numbered interfaces.

• OSPF over an IP unnumbered interface is supported.

• ISIS over an IP unnumbered interface is supported.

• BGP over a loopback interface with an IP unnumbered interface as an overlay interface is supported.

• The default and non-default VRF is supported by IP unnumbered interfaces.

• The switch has a limit of 16 user-defined MAC addresses (MEv6/static). Configuring beyond this limit might result in issues documented in CSCux84428.
If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Default Settings

The following table lists the default settings for Layer 3 interface parameters.

**Table 8: Default Layer 3 Interface Parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin state</td>
<td>Shut</td>
</tr>
</tbody>
</table>

Configuring Layer 3 Interfaces

Configuring a Routed Interface

You can configure any Ethernet port as a routed interface.

**SUMMARY STEPS**

1. configure terminal
2. interface ethernet slot/port
3. no switchport
4. [ip address ip-address/length | ipv6 address ipv6-address/length]
5. show interfaces
6. no shutdown
7. copy running-config startup-config

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td></td>
<td>switch(config)#</td>
</tr>
<tr>
<td>Step 2</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>switch(config)# interface ethernet slot/port</td>
</tr>
<tr>
<td></td>
<td>switch(config-if)#</td>
</tr>
</tbody>
</table>
### Configuring Layer 3 Interfaces

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 3** | Configures the interface as a Layer 3 interface.  
no switchport  
**Example:**  
switch(config-if)# no switchport |
| **Step 4** | • Configures an IP address for this interface. See the [Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide](https://www.cisco.com/c/en/us/support/docs/switches/routers/9000-series-nx-os-unicast-routing.html) for more information about IP addresses.  
[ip address ip-address/length | ipv6 address ipv6-address/length]  
**Example:**  
switch(config-if)# ip address 192.0.2.1/8  
switch(config-if)# ipv6 address 2001:0DB8::1/8 |
| **Step 5** | (Optional) Displays the Layer 3 interface statistics.  
show interfaces  
**Example:**  
switch(config-if)# show interfaces ethernet 2/1 |
| **Step 6** | (Optional) Clears the errors on the interfaces where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.  
no shutdown  
**Example:**  
switch#  
switch(config-if)# int e2/1  
switch(config-if)# no shutdown |
| **Step 7** | (Optional) Saves the configuration change.  
copy running-config startup-config  
**Example:**  
switch(config)# copy running-config startup-config |

### Example

- Use the `medium` command to set the interface medium to either point to point or broadcast.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| medium {broadcast | p2p}  
**Example:**  
switch(config-if)# medium p2p medium p2p | Configures the interface medium as either point to point or broadcast. |

### Note

The default setting is `broadcast`, and this setting does not appear in any of the `show` commands. However, if you do change the setting to `p2p`, you will see this setting when you enter the `show running config` command.

- Use the `switchport` command to convert a Layer 3 interface into a Layer 2 interface.
Configure the interface as a Layer 2 interface and deletes any configuration specific to Layer 3 on this interface.

Example:

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

This example shows how to configure a routed interface:

```
switch# configure terminal
switch(config)# interface ethernet 2/1
switch(config-if)# no switchport
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# copy running-config startup-config
```

The default setting for interfaces is routed. If you want to configure an interface for Layer 2, enter the `switchport` command. Then, if you change a Layer 2 interface to a routed interface, enter the `no switchport` command.

### Configuring a Subinterface on a Routed Interface

You can configure one or more subinterfaces on a routed interface made from routed interfaces.

**Before you begin**

Configure the parent interface as a routed interface.

See the “Configuring a Routed Interface” section.

**SUMMARY STEPS**

1. `configure terminal`
2. `interface ethernet slot/port.number`
3. `[ip address ip-address/length | ipv6 address ipv6-address/length]`
4. `encapsulation dot1Q vlan-id`
5. `show interfaces`
6. `copy running-config startup-config`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**
| configure terminal |
| Example:
| switch# configure terminal
| switch(config)# |
| **Step 2**
| interface ethernet slot/port.number |
| Example:
| switch(config)# interface ethernet 2/1.1
| switch(config-subif)# |
| Note | A subinterface greater than 511 is not supported on physical interfaces. |
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 3</td>
<td>[ip address ip-address/length</td>
<td>ipv6 address ipv6-address/length]</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-subif)# ip address 192.0.2.1/8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-subif)# ipv6 address 2001:0DB8::1/8</td>
<td>• Configures an IPv6 address for this subinterface. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information on IPv6 addresses.</td>
</tr>
<tr>
<td>Step 4</td>
<td>encapsulation dot1Q vlan-id</td>
<td>Configures IEEE 802.1Q VLAN encapsulation on the subinterface. The range is from 2 to 4093.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-subif)# encapsulation dot1Q 33</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>show interfaces</td>
<td>(Optional) Displays the Layer 3 interface statistics.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-subif)# show interfaces ethernet 2/1.1</td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td>copy running-config startup-config</td>
<td>(Optional) Saves the configuration change.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)# copy running-config startup-config</td>
<td></td>
</tr>
</tbody>
</table>

**Example**

- This example shows how to create a subinterface:
  ```
  switch# configure terminal
  switch(config)# interface ethernet 2/1.1
  switch(config-if)# ip address 192.0.2.1/8
  switch(config-if)# encapsulation dot1Q 33
  switch(config-if)# copy running-config startup-config
  ```

- The output of the `show interface eth` command is enhanced for the subinterfaces as shown in the following:
  ```
  switch# show interface ethernet 1/2.1
  Ethernet1/2.1 is down (Parent Interface Admin down)
  admin state is down, Dedicated Interface, [parent interface is Ethernet1/2]
  Hardware: 40000 Ethernet, address: 0023.ac67.9bc1 (bia 4055.3926.61d4)
  Internet Address is 10.10.10.1/24
  MTU 1500 bytes, BW 40000000 Kbit, DLY 10 usec
  reliability 255/255, txload 1/255, rxload 1/255
  Auto-mdix is turned off
  EtherType is 0x8100
  L3 in Switched:
    ucast: 0 pkts, 0 bytes - mcast: 0 pkts, 0 bytes
  L3 out Switched:
    ucast: 0 pkts, 0 bytes - mcast: 0 pkts, 0 bytes
  ```
Configuring a Subinterface on a Port-Channel Interface

You can configure one or more subinterfaces on a port-channel interface.

**Note**

Subinterfaces on a port-channel interface do not support multicast routing, router ACLs, QoS, policy-based routing (PBR), SPAN, or ERSPAN.

**Before you begin**

Configure the parent interface as a port-channel interface.

See the “Configuring Port Channels” chapter.

**SUMMARY STEPS**

1. configure terminal
2. interface port-channel channel-id.number
3. [ip address ip-address/length | ipv6 address ipv6-address/length]
4. encapsulation dot1Q vlan-id
5. show interfaces
6. copy running-config startup-config

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config) #</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>interface port-channel channel-id.number</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config) #</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>[ip address ip-address/length</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config-subif) #</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config-subif) #</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>encapsulation dot1Q vlan-id</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>

- Configures an IP address for this subinterface. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information on IP addresses.
- Configures an IPv6 address for this subinterface. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information on IPv6 addresses.
### Configuring a VLAN Interface

You can create VLAN interfaces to provide inter-VLAN routing.

**SUMMARY STEPS**

1. configure terminal
2. feature interface-vlan
3. interface vlan number
4. ip address ip-address/length | ipv6 address ipv6-address/length
5. show interface vlan number
6. no shutdown
7. copy running-config startup-config

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>switch(config)# interface port-channel</td>
<td></td>
</tr>
<tr>
<td>115.3</td>
<td></td>
</tr>
<tr>
<td>switch(config)# ip address 141.143.101.2/24</td>
<td></td>
</tr>
<tr>
<td>switch(config)# encapsulation dot1q 3</td>
<td></td>
</tr>
<tr>
<td>switch(config)# copy running-config startup-config</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
</tr>
<tr>
<td>2</td>
<td>feature interface-vlan</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>switch(config)# feature interface-vlan</td>
</tr>
<tr>
<td>3</td>
<td>interface vlan number</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>switch(config)# interface vlan 10</td>
</tr>
<tr>
<td></td>
<td>switch(config-if)#</td>
</tr>
<tr>
<td>4</td>
<td>[ip address ip-address/length</td>
</tr>
<tr>
<td></td>
<td>ipv6 address ipv6-address/length]</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# ip address 192.0.2.1/8</td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# ipv6 address 2001:0DB8::1/8</td>
</tr>
<tr>
<td>5</td>
<td>show interface vlan number</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# show interface vlan 10</td>
</tr>
<tr>
<td>6</td>
<td>no shutdown</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>switch(config)# int e3/1</td>
</tr>
<tr>
<td></td>
<td>switch(config)# no shutdown</td>
</tr>
<tr>
<td>7</td>
<td>copy running-config startup-config</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# copy running-config startup-config</td>
</tr>
</tbody>
</table>

**Example**

This example shows how to create a VLAN interface:

```
switch# configure terminal
switch(config)# feature interface-vlan
switch(config)# interface vlan 10
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# copy running-config startup-config
```
Enabling Layer 3 Retention During VRF Membership Change

The following steps enable the retention of the Layer 3 configuration when changing the VRF membership on the interface.

**SUMMARY STEPS**

1. configure terminal
2. system vrf-member-change retain-l3-config

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>configure terminal</td>
<td></td>
</tr>
</tbody>
</table>
| Example:
switch# configure terminal
switch(config)#                                         |                                                         |
| **Step 2**                                             | Enables Layer 3 configuration retention during VRF membership change. |
| system vrf-member-change retain-l3-config              |                                                         |
| Example:
switch(config)# system vrf-member-change retain-l3-config |                                                         |
| Warning: Will retain L3 configuration when vrf member change on interface. | To disable the retention of the Layer 3 configuration, use the no system vrf-member-change retain-l3-config command. |

Configuring a Loopback Interface

You can configure a loopback interface to create a virtual interface that is always up.

**Before you begin**

Ensure that the IP address of the loopback interface is unique across all routers on the network.

**SUMMARY STEPS**

1. configure terminal
2. interface loopback instance
3. [ip address ip-address/length | ipv6 address ipv6-address/length]
4. show interface loopback instance
5. copy running-config startup-config
### Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**  
  configure terminal  
  Example:  
  `switch# configure terminal`  
  `switch(config)#` | Enters configuration mode.                                               |
| **Step 2**  
  interface loopback *instance*  
  Example:  
  `switch(config)# interface loopback 0`  
  `switch(config-if)#` | Creates a loopback interface. The range is from 0 to 1023.               |
| **Step 3**  
  [ip address ip-address/length | ipv6 address ipv6-address/length]  
  Example:  
  `switch(config-if)# ip address 192.0.2.1/8`  
  `switch(config-if)# ipv6 address 2001:0DB8::1/8` | • Configures an IP address for this interface. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information about IP addresses.  
  • Configures an IPv6 address for this interface. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information about IPv6 addresses. |
| **Step 4**  
  show interface loopback *instance*  
  Example:  
  `switch(config-if)# show interface loopback 0` | (Optional) Displays the loopback interface statistics.                   |
| **Step 5**  
  copy running-config startup-config  
  Example:  
  `switch(config-if)# copy running-config startup-config` | (Optional) Saves the configuration change.                               |

**Example**

This example shows how to create a loopback interface:

```
switch# configure terminal
switch(config)# interface loopback 0
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# ipv6 address 2001:0DB8::1/8
switch(config-if)# copy running-config startup-config
```

### Configuring IP Unnumbered on an Ethernet Interface

You can configure the IP unnumbered feature on an ethernet interface.

**Summary Steps**

1. configure terminal
2. interface ethernet *slot/port*
3. medium p2p  
4. ip unnumbered type number

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: switch# configure terminal switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface ethernet slot/port</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td>Example: switch(config)# interface ethernet 1/1 switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> medium p2p</td>
<td>Configures the interface medium as point to point.</td>
</tr>
<tr>
<td>Example: switch(config-if)# medium p2p</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> ip unnumbered type number</td>
<td>Enables IP processing on an interface without assigning an explicit IP address to the interface.</td>
</tr>
<tr>
<td>Example: switch(config-if)# ip unnumbered loopback 100</td>
<td>type and number specify another interface on which the router has an assigned IP address. The interface specified cannot be another unnumbered interface.</td>
</tr>
<tr>
<td>Note</td>
<td>type is limited to loopback.</td>
</tr>
</tbody>
</table>

**Configuring OSPF for an IP Unnumbered Interface**

You can configure OSPF for an IP unnumbered loopback interface.

**SUMMARY STEPS**

1. configure terminal  
2. interface ethernet slot/port  
3. encapsulation dot1Q vlan-id  
4. medium p2p  
5. ip unnumbered type number  
6. (Optional) ip ospf authentication  
7. (Optional) ip ospf authentication-key password  
8. ip router ospf instance area area-number  
9. no shutdown  
10. interface loopback instance  
11. ip address ip-address/length  
12. ip router ospf instance area area-number
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config)#</code></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>interface ethernet slot/port</code></td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config)# interface ethernet 1/20.1</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-if)#</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>encapsulation dot1Q vlan-id</code></td>
<td>Configures IEEE 802.1Q VLAN encapsulation on the subinterface. The range is from 2 to 4093.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-if)# encapsulation dot1Q 100</code></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>medium p2p</code></td>
<td>Configures the interface medium as point to point.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-if)# medium p2p</code></td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td><code>ip unnumbered type number</code></td>
<td>Enables IP processing on an interface without assigning an explicit IP address to the interface. type and number specify another interface on which the router has an assigned IP address. The interface specified cannot be another unnumbered interface.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-if)# ip unnumbered loopback 101</code></td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td><em>(Optional)</em> <code>ip ospf authentication</code></td>
<td>Specifies the authentication type for interface.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-if)# ip ospf authentication</code></td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td><em>(Optional)</em> <code>ip ospf authentication-key password</code></td>
<td>Specifies the authentication password for OSPF authentication.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-if)# ip ospf authentication 3 b7bdf15f62bbd250</code></td>
<td></td>
</tr>
<tr>
<td>Step 8</td>
<td><code>ip router ospf instance area area-number</code></td>
<td>Configures routing process for IP on an interface and specifies an area.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-if)# ip router ospf 100 area 0.0.0.1</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td>The <strong>ip router ospf</strong> command is required for both the unnumbered and the numbered interface.</td>
</tr>
<tr>
<td>Step 9</td>
<td><code>no shutdown</code></td>
<td>Brings up the interface (administratively).</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-if)# no shutdown</code></td>
<td></td>
</tr>
</tbody>
</table>
### Configuring IS-IS for an IP Unnumbered Interface

You can configure IS-IS for an IP unnumbered loopback interface.

#### SUMMARY STEPS

1. configure terminal
2. feature isis
3. router isis area-tag
4. net network-entity-title
5. end
6. interface ethernet slot/port
7. encapsulation dot1Q vlan-id
8. medium p2p
9. ip unnumbered type number
10. ip router isis area-tag
11. no shutdown

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>feature isis</td>
<td>Enables IS-IS.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Switch(config)# feature isis</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------</td>
<td>----------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3</td>
<td><code>router isis area-tag</code></td>
<td>Assigns a tag to an IS-IS process and enters router configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Switch(config)# router isis 100</code></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><code>net network-entity-title</code></td>
<td>Configures the network entity title (NET) on the device.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Switch(config-router)# net 49.0001.0100.0100.1001.00</code></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><code>end</code></td>
<td>Exit router configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Switch(config-router)# end</code></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td><code>interface ethernet slot/port</code></td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config)# interface ethernet 1/20.1</code></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><code>encapsulation dot1Q vlan-id</code></td>
<td>Configures IEEE 802.1Q VLAN encapsulation on the subinterface. The range is from 2 to 4093.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-subif)# encapsulation dot1Q 100</code></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td><code>medium p2p</code></td>
<td>Configures the interface medium as point to point.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-subif)# medium p2p</code></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td><code>ip unnumbered type number</code></td>
<td>Enables IP processing on an interface without assigning an explicit IP address to the interface. <code>type</code> and <code>number</code> specify another interface on which the router has an assigned IP address. The interface specified cannot be another unnumbered interface.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-if)# ip unnumbered loopback 101</code></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><code>ip router isis area-tag</code></td>
<td>Enables ISIS on the unnumbered interface.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-subif)# ip router isis 100</code></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td><code>no shutdown</code></td>
<td>Brings up the interface (administratively).</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-subif)# no shutdown</code></td>
<td></td>
</tr>
</tbody>
</table>

### Configuring PBR on SVI on the Gateway

This procedure configures PBR on the primary SVI interface in the gateway.
Steps 2 through 6 are needed if you want to configure a PBR policy on the unnumbered Primary/Secondary VLAN interfaces. This is not mandatory for IP unnumbered on the SVI feature.

### SUMMARY STEPS

1. configure terminal
2. ip access-list list-name
3. permit tcp host ipaddr host ipaddr eq port-number
4. exit
5. route-map route-map-name
6. match ip address access-list-name
7. set ip next-hop addr1
8. exit
9. interface vlan vlan-id
10. ip address ip-addr
11. no ip redirects
12. (Optional) ip policy route-map pbr-sample
13. exit
14. hsrp version 2
15. hsrpgroup-num
16. name name-val
17. ip ip-addr
18. no shutdown

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td>Example:</td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>ip access-list list-name</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config)# ip access-list pbr-sample</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>permit tcp host ipaddr host ipaddr eq port-number</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config-acl)# permit tcp host 10.1.1.1 host 192.168.2.1 eq 80</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>exit</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config-acl)# exit</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Create a route-map or enter route-map command mode.</td>
</tr>
<tr>
<td>route-map route-map-name</td>
<td>Example: switch(config)# route-map pbr-sample</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Match values from the routing table.</td>
</tr>
<tr>
<td>match ip address access-list-name</td>
<td>Example: switch(config-route-map)# match ip address pbr-sample</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Set IP address of the next hop.</td>
</tr>
<tr>
<td>set ip next-hop addr1</td>
<td>Example: switch(config-route-map)# set ip next-hop 192.168.1.1</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Exit command mode.</td>
</tr>
<tr>
<td>exit</td>
<td>Example: switch(config-route-map)# exit</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>Creates a VLAN interface and enters interface configuration mode. The range is from 1 and 4094. This is the primary VLAN.</td>
</tr>
<tr>
<td>interface vlan vlan-id</td>
<td>Example: switch(config)# interface vlan 2003</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>Configures an IP address for the interface.</td>
</tr>
<tr>
<td>ip address ip-addr</td>
<td>Example: switch(config-if)# ip address 10.0.0.1/8</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>Needs to be configured on all unnumbered primary and secondary VLAN interfaces.</td>
</tr>
<tr>
<td>no ip redirects</td>
<td>Example: switch(config-if)# no ip redirects</td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td>Enter this command if you want to apply a PBR policy on the unnumbered Primary/Secondary VLAN interface.</td>
</tr>
<tr>
<td>Example: switch(config-if)# ip policy route-map pbr-sample</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td>Exit command mode.</td>
</tr>
<tr>
<td>exit</td>
<td>Example: switch(config-if)# exit</td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td>Set the HSRP version.</td>
</tr>
<tr>
<td>hsrp version 2</td>
<td>Example: switch(config-if)# hsrp version 2</td>
</tr>
<tr>
<td><strong>Step 15</strong></td>
<td>Set the HSRP group number.</td>
</tr>
<tr>
<td>hsrp-group-num</td>
<td>Example: switch(config-if)# hsrp 200</td>
</tr>
</tbody>
</table>
### Purpose

**Command or Action**

**Purpose**

<table>
<thead>
<tr>
<th>Step 16</th>
<th>name name-val</th>
<th>Configure the redundancy name string.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>switch(config-if-hsrp)# name master</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 17</th>
<th>ip ip-addr</th>
<th>Configures an IP address.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>switch(config-if-hsrp)# ip 10.0.0.100</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 18</th>
<th>no shutdown</th>
<th>Negates shutdown.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>switch(config-if-hsrp)# no shutdown</code></td>
<td></td>
</tr>
</tbody>
</table>

---

### Configuring IP Unnumbered on SVI Secondary VLAN on the Gateway

This procedure configures IP unnumbered on the secondary SVI in the gateway.

**SUMMARY STEPS**

1. configure terminal
2. interface vlan vlan-list
3. ip unnumbered vlan primary-vlan-id
4. (Optional) ip policy route-map pbr-sample
5. no ip redirects
6. hsrp version 2
7. hsrp group-num
8. follow name
9. ip ip-addr
10. no shutdown

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td>Example:</td>
<td><code>switch# configure terminal</code></td>
</tr>
</tbody>
</table>

| **Step 2** | interface vlan vlan-list |
| Example: | `switch(config)# interface vlan 2001` |

| **Step 3** | ip unnumbered vlan primary-vlan-id |
| Example: | `switch(config-if)# ip unnumbered vlan 2003` |
### Purpose

**Command or Action**

**Step 4**

(Optional) `ip policy route-map pbr-sample`  
**Example:**

```
switch(config-if)# ip policy route-map pbr-sample
```

Enter this command if you want to apply a PBR policy on the unnumbered Primary/Secondary VLAN interface.

**Step 5**

`no ip redirects`  
**Example:**

```
switch(config-if)# no ip redirects
```

Needs to be configured on all unnumbered primary and secondary VLAN interfaces.

**Step 6**

`hsrp version 2`  
**Example:**

```
switch(config-if)# hsrp version 2
```

Set the HSRP version.

**Step 7**

`hsrp group-num`  
**Example:**

```
switch(config-if)# hsrp 200
```

Set the HSRP group number.

**Step 8**

`follow name`  
**Example:**

```
switch(config-if-hsrp)# follow master
```

Configure the group to be followed.

**Step 9**

`ip ip-addr`  
**Example:**

```
switch(config-if-hsrp)# ip 10.0.0.100
```

Enters HSRP IPv4 and sets the virtual IP address.

**Step 10**

`no shutdown`  
**Example:**

```
switch(config-if-hsrp)# no shutdown
```

Negate shutdown.

---

**Assigning an Interface to a VRF**

You can add a Layer 3 interface to a VRF.

**SUMMARY STEPS**

1. `configure terminal`  
2. `interface interface-type number`  
3. `vrf member vrf-name`  
4. `ip address ip-prefix/length`  
5. `show vrf [vrf-name] interface interface-type number`  
6. `copy running-config startup-config`
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> switch# configure terminal switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface interface-type number</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> switch(config)# interface loopback 0 switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> vrf member vrf-name</td>
<td>Adds this interface to a VRF.</td>
</tr>
<tr>
<td><strong>Example:</strong> switch(config-if)# vrf member RemoteOfficeVRF</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> ip address ip-prefix/length</td>
<td>Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.</td>
</tr>
<tr>
<td><strong>Example:</strong> switch(config-if)# ip address 192.0.2.1/16</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> show vrf [vrf-name] interface interface-type number</td>
<td>(Optional) Displays VRF information.</td>
</tr>
<tr>
<td><strong>Example:</strong> switch(config-vrf)# show vrf Enterprise interface loopback 0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> copy running-config startup-config</td>
<td>(Optional) Saves the configuration change.</td>
</tr>
<tr>
<td><strong>Example:</strong> switch(config-if)# copy running-config startup-config</td>
<td></td>
</tr>
</tbody>
</table>

**Example**

This example shows how to add a Layer 3 interface to the VRF:

```
switch# configure terminal
switch(config)# interface loopback 0
switch(config-if)# vrf member RemoteOfficeVRF
switch(config-if)# ip address 209.0.2.1/16
switch(config-if)# copy running-config startup-config
```

### Configuring a MAC-Embedded IPv6 Address

You can configure a MAC-embedded IPv6 (MEv6) address.

**SUMMARY STEPS**

1. configure terminal
2. `interface type slot/port`
3. `no switchport`
4. `mac-address ipv6-extract`
5. `ipv6 address ip-address/length`
6. `ipv6 nd mac-extract [exclude nud-phase]`
7. (Optional) `show ipv6 icmp interface type slot/port`
8. (Optional) `copy running-config startup-config`

### Detailed Steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Step 1 | `configure terminal`  
*Example:*  
`switch# configure terminal  
switch(config)#` | Enters global configuration mode. |
| Step 2 | `interface type slot/port`  
*Example:*  
`switch(config)# interface ethernet 1/3  
switch(config-if)#` | Enters the interface configuration mode for the specified interface. |
| Step 3 | `no switchport`  
*Example:*  
`switch(config-if)# no switchport` | Configures the interface as a Layer 3 interface and deletes any configuration specific to Layer 2 on this interface.  
**Note** To convert a Layer 3 interface back into a Layer 2 interface, use the `switchport` command. |
| Step 4 | `mac-address ipv6-extract`  
*Example:*  
`switch(config-if)# mac-address ipv6-extract` | Extracts the MAC address embedded in the IPv6 address configured on the interface.  
**Note** The MEv6 configuration is currently not supported with the EUI-64 format of IPv6 address. |
| Step 5 | `ipv6 address ip-address/length`  
*Example:*  
`switch(config-if)# ipv6 address 2002:1::10/64` | Configures an IPv6 address for this interface. |
| Step 6 | `ipv6 nd mac-extract [exclude nud-phase]`  
*Example:*  
`switch(config-if)# ipv6 nd mac-extract` | Extracts the next-hop MAC address embedded in a next-hop IPv6 address.  
The `exclude nud-phase` option blocks packets during the ND phase only. When the `exclude nud-phase` option is not specified, packets are blocked during both ND and neighbor unreachability detection (NUD) phases. |
| Step 7 | (Optional) `show ipv6 icmp interface type slot/port`  
*Example:*  
`switch(config-if)# show ipv6 icmp interface ethernet 1/3` | Displays IPv6 Internet Control Message Protocol version 6 (ICMPv6) interface information. |
### Command or Action

| Step 8 | (Optional) copy running-config startup-config |

**Example:**

```
switch(config-if)# copy running-config startup-config
```

**Purpose:** Copies the running configuration to the startup configuration.

---

### Example

This example shows how to configure a MAC-embedded IPv6 address with ND mac-extract enabled:

```bash
switch# configure terminal
switch(config)# interface ethernet 1/3
switch(config-if)# no switchport
switch(config-if)# mac-address ipv6-extract
switch(config-if)# ipv6 address 2002:1::10/64
switch(config-if)# ipv6 nd mac-extract
switch(config-if)# show ipv6 icmp interface ethernet 1/3
```

ICMPv6 Interfaces for VRF "default"

<table>
<thead>
<tr>
<th>Ethernet1/3, Interface status: protocol-up/link-up/admin-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv6 address: 2002:1::10</td>
</tr>
<tr>
<td>IPv6 subnet: 2002:1::/64</td>
</tr>
<tr>
<td>ND mac-extract : Enabled</td>
</tr>
<tr>
<td>ICMPv6 active timers:</td>
</tr>
<tr>
<td>Last Neighbor-Solicitation sent: 00:01:39</td>
</tr>
<tr>
<td>Last Neighbor-Advertisement sent: 00:01:40</td>
</tr>
<tr>
<td>Last Router-Advertisement sent: 00:01:41</td>
</tr>
<tr>
<td>Next Router-Advertisement sent in: 00:03:34</td>
</tr>
<tr>
<td>Router-Advertisement parameters:</td>
</tr>
<tr>
<td>Periodic interval: 200 to 600 seconds</td>
</tr>
<tr>
<td>Send &quot;Managed Address Configuration&quot; flag: false</td>
</tr>
<tr>
<td>Send &quot;Other Stateful Configuration&quot; flag: false</td>
</tr>
<tr>
<td>Send &quot;Current Hop Limit&quot; field: 64</td>
</tr>
<tr>
<td>Send &quot;MTU&quot; option value: 1500</td>
</tr>
<tr>
<td>Send &quot;Router Lifetime&quot; field: 1800 secs</td>
</tr>
<tr>
<td>Send &quot;Reachable Time&quot; field: 0 ms</td>
</tr>
<tr>
<td>Send &quot;Retrans Timer&quot; field: 0 ms</td>
</tr>
<tr>
<td>Suppress RA: Disabled</td>
</tr>
<tr>
<td>Suppress MTU in RA: Disabled</td>
</tr>
<tr>
<td>Neighbor-Solicitation parameters:</td>
</tr>
<tr>
<td>NS retransmit interval: 1000 ms</td>
</tr>
<tr>
<td>ICMPv6 error message parameters:</td>
</tr>
<tr>
<td>Send redirects: true</td>
</tr>
<tr>
<td>Send unreachables: false</td>
</tr>
<tr>
<td>ICMPv6-nd Statistics (sent/received):</td>
</tr>
<tr>
<td>RAs: 3/0, RSs: 0/0, NAs: 2/0, NSs: 7/0, RDs: 0/0</td>
</tr>
<tr>
<td>Interface statistics last reset: never</td>
</tr>
</tbody>
</table>

---

This example shows how to configure a MAC-embedded IPv6 address with ND mac-extract (excluding NUD phase) enabled:

```bash
switch# configure terminal
switch(config)# interface ethernet 1/5
switch(config-if)# no switchport
switch(config-if)# mac-address ipv6-extract
switch(config-if)# ipv6 address 2002:2::10/64
switch(config-if)# ipv6 nd mac-extract exclude nud-phase
switch(config-if)# show ipv6 icmp interface ethernet 1/5
```

---

_Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide, Release 9.2(x)_

117
ICMPv6 Interfaces for VRF "default"
Ethernet1/5, Interface status: protocol-up/link-up/admin-up
IPv6 address: 2002:2::10
IPv6 subnet: 2002:2::/64
IPv6 interface DAD state: VALID
ND mac-extract : Enabled (Excluding NUD Phase)
ICMPv6 active timers:
Last Neighbor-Solicitation sent: 00:06:45
Last Neighbor-Advertisement sent: 00:06:46
Last Router-Advertisement sent: 00:02:18
Next Router-Advertisement sent in: 00:02:24
Router-Advertisement parameters:
  Periodic interval: 200 to 600 seconds
  Send "Managed Address Configuration" flag: false
  Send "Other Stateful Configuration" flag: false
  Send "Current Hop Limit" field: 64
  Send "MTU" option value: 1500
  Send "Router Lifetime" field: 1800 secs
  Send "Reachable Time" field: 0 ms
  Send "Retrans Timer" field: 0 ms
  Suppress RA: Disabled
  Suppress MTU in RA: Disabled
Neighbor-Solicitation parameters:
  NS retransmit interval: 1000 ms
ICMPv6 error message parameters:
  Send redirects: true
  Send unreachables: false
ICMPv6-nd Statisitcs (sent/received):
  RAs: 6/0, RSs: 0/0, NAs: 2/0, NSs: 7/0, RDs: 0/0
Interface statistics last reset: never

Configuring a DHCP Client on an Interface

You can configure the DHCP client on an SVI, a management interface, or a physical Ethernet interface for IPv4 or IPv6 address

SUMMARY STEPS

1. switch# configure terminal
2. switch(config)# interface ethernet type slot/port | mgmt mgmt-interface-number | vlan vlan id
3. switch(config-if)# [no] ipv6 address use-link-local-only
4. switch(config-if)# [no] [ip | ipv6] address dhcp
5. (Optional) switch(config)# copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# interface ethernet type slot/port</td>
<td>mgmt mgmt-interface-number</td>
</tr>
<tr>
<td><strong>Step 3</strong> switch(config-if)# [no] ipv6 address use-link-local-only</td>
<td>Prepares for request to the DHCP server.</td>
</tr>
</tbody>
</table>
### Configuring Layer 3 Interfaces

#### Command or Action

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Command or Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Note</strong></td>
<td>This command is only required for an IPv6 address.</td>
</tr>
</tbody>
</table>
| **Step 4** | `switch(config-if)# [no] [ip | ipv6] address dhcp`
| | Requests the DHCP server for an IPv4 or IPv6 address. The no form of this command removes any address that was acquired. |
| **Step 5** | `(Optional) switch(config)# copy running-config startup-config`
| | Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. |

### Example

This example shows how to configure the IP address of a DHCP client on an SVI:

```
switch# configure terminal
switch(config)# interface vlan 15
switch(config-if)# ip address dhcp
```

This example shows how to configure an IPv6 address of a DHCP client on a management interface:

```
switch# configure terminal
switch(config)# interface mgmt 0
switch(config-if)# ipv6 address use-link-local-only
switch(config-if)# ipv6 address dhcp
```

### Verifying the Layer 3 Interfaces Configuration

To display the Layer 3 configuration, perform one of the following tasks:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show interface ethernet slot/port</code></td>
<td>Displays the Layer 3 interface configuration, status, and counters (including the 5-minute exponentially decayed moving average of inbound and outbound packet and byte rates).</td>
</tr>
<tr>
<td><code>show interface ethernet slot/port brief</code></td>
<td>Displays the Layer 3 interface operational status.</td>
</tr>
<tr>
<td><code>show interface ethernet slot/port capabilities</code></td>
<td>Displays the Layer 3 interface capabilities, including port type, speed, and duplex.</td>
</tr>
<tr>
<td><code>show interface ethernet slot/port description</code></td>
<td>Displays the Layer 3 interface description.</td>
</tr>
<tr>
<td><code>show interface ethernet slot/port status</code></td>
<td>Displays the Layer 3 interface administrative status, port mode, speed, and duplex.</td>
</tr>
</tbody>
</table>
Monitoring the Layer 3 Interfaces

Use the following commands to display Layer 3 statistics:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>load-interval</strong> `interval seconds {1</td>
<td>2</td>
</tr>
<tr>
<td><strong>show interface ethernet slot/port counters</strong></td>
<td>Displays the Layer 3 interface statistics (unicast, multicast, and broadcast).</td>
</tr>
</tbody>
</table>
### Configuration Examples for Layer 3 Interfaces

This example shows how to configure Ethernet subinterfaces:

```plaintext
interface ethernet 2/1.10
description Layer 3
ip address 192.0.2.1/8
```

This example shows how to configure a loopback interface:

```plaintext
interface loopback 3
ip address 192.0.2.2/32
```

---

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>show interface ethernet slot/port counters brief</td>
<td>Displays the Layer 3 interface input and output counters.</td>
</tr>
<tr>
<td>show interface ethernet errors slot/port detailed [all]</td>
<td>Displays the Layer 3 interface statistics. You can optionally include all 32-bit and 64-bit packet and byte counters (including errors).</td>
</tr>
<tr>
<td>show interface ethernet errors slot/port counters errors</td>
<td>Displays the Layer 3 interface input and output errors.</td>
</tr>
<tr>
<td>show interface ethernet errors slot/port counters snmp</td>
<td>Displays the Layer 3 interface counters reported by SNMP MIBs.</td>
</tr>
<tr>
<td>show interface ethernet slot/port number counters</td>
<td>Displays the subinterface statistics (unicast, multicast, and broadcast).</td>
</tr>
<tr>
<td>show interface port-channel channel-id.number counters</td>
<td>Displays the port-channel subinterface statistics (unicast, multicast, and broadcast).</td>
</tr>
<tr>
<td>show interface loopback number counters</td>
<td>Displays the loopback interface input and output counters (unicast, multicast, and broadcast).</td>
</tr>
<tr>
<td>show interface loopback number detailed [all]</td>
<td>Displays the loopback interface statistics. You can optionally include all 32-bit and 64-bit packet and byte counters (including errors).</td>
</tr>
<tr>
<td>show interface loopback number counters errors</td>
<td>Displays the loopback interface input and output errors.</td>
</tr>
<tr>
<td>show interface vlan number counters</td>
<td>Displays the VLAN interface input and output counters (unicast, multicast, and broadcast).</td>
</tr>
<tr>
<td>show interface vlan number counters detailed [all]</td>
<td>Displays the VLAN interface statistics. You can optionally include all Layer 3 packet and byte counters (unicast and multicast).</td>
</tr>
<tr>
<td>show interface vlan number counters snmp</td>
<td>Displays the VLAN interface counters reported by SNMP MIBs.</td>
</tr>
</tbody>
</table>
Example of Changing VRF Membership for an Interface

- Enable Layer 3 configuration retention when changing VRF membership.

  switch# configure terminal
  switch(config)# system vrf-member-change retain-l3-config

  Warning: Will retain L3 configuration when vrf member change on interface.

- Verify Layer 3 retention.

  switch# show running-config | include vrf-member-change
  system vrf-member-change retain-l3-config

- Configure the SVI interface with Layer 3 configuration as VRF "blue".

  switch# configure terminal
  switch(config)# show running-config interface vlan 2002

  interface Vlan2002
  description TESTSVI
  no shutdown
  mtu 9192
  vrf member blue
  no ip redirects
  ip address 192.168.211.2/27
  ipv6 address 2620:10d:c041:12::2/64
  ipv6 link-local fe80::1
  ip router ospf 1 area 0.0.0.0
  ipv6 router ospfv3 1 area 0.0.0.0
  hsrp version 2
  hsrp 2002
  preempt delay minimum 300 reload 600
  priority 110 forwarding-threshold lower 1 upper 110
  ip 192.168.211.1
  hsrp 2002 ipv6
  preempt delay minimum 300 reload 600
  priority 110 forwarding-threshold lower 1 upper 110
  ip 2620:10d:c041:12::1

- Change the SVI interface VRF to "red".

  switch# configure terminal
  Enter configuration commands, one per line. End with CNTL/Z.
  switch(config)# interface vlan 2002
  switch(config-if)# vrf member red

  Warning: Retain-L3-config is on, deleted and re-added L3 config on interface Vlan2002

- Verify SVI interface after VRF change.

  switch# configure terminal
  switch(config)# show running-config interface vlan 2002

  interface Vlan2002
  description TESTSVI
  no shutdown
  mtu 9192
vrf member red
no ip redirects
ip address 192.168.211.2/27
ipv6 address 2620:10d:c041:12::2/64
ipv6 link-local fe80::1
ip router ospf 1 area 0.0.0.0
ipv6 router ospfv3 1 area 0.0.0.0
hsrp version 2
hsrp 2002
preempt delay minimum 300 reload 600
priority 110 forwarding-threshold lower 1 upper 110
ip 192.168.211.1
hsrp 2002 ipv6
preempt delay minimum 300 reload 600
priority 110 forwarding-threshold lower 1 upper 110
ip 2620:10d:c041:12::1

- When changing the VRF, the Layer 3 configuration retention affects:
  - Physical Interface
  - Loopback Interface
  - SVI Interface
  - Sub-interface
  - Tunnel Interface
  - Port-Channel

- When changing the VRF, the existing Layer 3 configuration is deleted and reapplied. All routing protocols, such as OSPF/ISIS/EIGRP/HSRP, go down in the old VRF and come up in the new VRF.
- Direct/Local IPv4/IPv6 addresses are removed from the old VRF and installed in the new VRF.
- Some traffic loss might occur during the VRF change.

## Related Documents

<table>
<thead>
<tr>
<th>Related Documents</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide</td>
</tr>
<tr>
<td>VLANs</td>
<td>Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide</td>
</tr>
</tbody>
</table>
About BFD

BFD is a detection protocol designed to provide fast forwarding-path failure detection times for media types, encapsulations, topologies, and routing protocols. You can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different protocol hello mechanisms. BFD makes network profiling and planning easier and reconvergence time consistent and predictable.

BFD provides subsecond failure detection between two adjacent devices and can be less CPU-intensive than protocol hello messages because some of the BFD load can be distributed onto the data plane on supported modules.

Asynchronous Mode

Cisco NX-OS supports the BFD asynchronous mode, which sends BFD control packets between two adjacent devices to activate and maintain BFD neighbor sessions between the devices. You configure BFD on both devices (or BFD neighbors). Once BFD has been enabled on the interfaces and on the appropriate protocols, Cisco NX-OS creates a BFD session, negotiates BFD session parameters, and begins to send BFD control packets to each BFD neighbor at the negotiated interval. The BFD session parameters include the following:

- Desired minimum transmit interval—The interval at which this device wants to send BFD hello messages.
• Required minimum receive interval—The minimum interval at which this device can accept BFD hello messages from another BFD device.

• Detect multiplier—The number of missing BFD hello messages from another BFD device before this local device detects a fault in the forwarding path.

The following figure shows how a BFD session is established. The figure shows a simple network with two routers running Open Shortest Path First (OSPF) and BFD. When OSPF discovers a neighbor (1), it sends a request to the local BFD process to initiate a BFD neighbor session with the OSPF neighbor router (2). The BFD neighbor session with the OSPF neighbor router is now established (3).

**Figure 6: Establishing a BFD Neighbor Relationship**

Once a BFD session has been established and timer negotiations are complete, BFD neighbors send BFD control packets that act in the same manner as an IGP hello protocol to detect liveliness, except at a more accelerated rate. BFD detects a failure, but the protocol must take action to bypass a failed peer.

BFD sends a failure detection notice to the BFD-enabled protocols when it detects a failure in the forwarding path. The local device can then initiate the protocol recalculation process and reduce the overall network convergence time.

The following figure shows what happens when a failure occurs in the network (1). The BFD neighbor session with the OSPF neighbor router is torn down (2). BFD notifies the local OSPF process that the BFD neighbor is no longer reachable (3). The local OSPF process tears down the OSPF neighbor relationship (4). If an alternative path is available, the routers immediately start converging on it.

**Note**

Note The BFD failure detection occurs in less than a second, which is much faster than OSPF Hello messages could detect the same failure.
Distributed Operation

Cisco NX-OS can distribute the BFD operation to compatible modules that support BFD. This process offloads the CPU load for BFD packet processing to the individual modules that connect to the BFD neighbors. All BFD session traffic occurs on the module CPU. The module informs the supervisor when a BFD failure is detected.

BFD Echo Function

The BFD echo function sends echo packets from the forwarding engine to the remote BFD neighbor. The BFD neighbor forwards the echo packet back along the same path in order to perform detection; the BFD neighbor does not participate in the actual forwarding of the echo packets. The echo function and the forwarding engine are responsible for the detection process. BFD can use the slow timer to slow down the asynchronous session when the echo function is enabled and reduce the number of BFD control packets that are sent between two BFD neighbors. Also, the forwarding engine tests the forwarding path on the remote (neighbor) system without involving the remote system, so there is less interpacket delay variability and faster failure detection times.

The echo function is without asymmetry when both BFD neighbors are running echo function.

Security

Cisco NX-OS uses the packet Time to Live (TTL) value to verify that the BFD packets came from an adjacent BFD peer. For all asynchronous and echo request packets, the BFD neighbor sets the TTL value to 255 and the local BFD process verifies the TTL value as 255 before processing the incoming packet. For the echo response packet, BFD sets the TTL value to 254.

You can configure SHA-1 authentication of BFD packets.

High Availability

BFD supports stateless restarts. After a reboot or supervisor switchover, Cisco NX-OS applies the running configuration and BFD immediately sends control packets to the BFD peers.

Figure 7: Tearing Down an OSPF Neighbor Relationship

![Figure showing OSPF Neighbor Relationship]

172.16.10.1

Router A

172.16.10.2

Router B

OSPF

OSPF neighbors

BFD neighbors

4

2

1

X

X

X

X
Virtualization Support

BFD supports virtual routing and forwarding instances (VRFs). VRFs exist within virtual device contexts (VDCs). By default, Cisco NX-OS places you in the default VDC and default VRF.

Licensing Requirements for BFD

The following table shows the licensing requirements for this feature:

<table>
<thead>
<tr>
<th>Product</th>
<th>License Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco NX-OS</td>
<td>BFD requires no license. Any feature not included in a license package is bundled with the NX-OS image and is provided at no extra charge to you.</td>
</tr>
</tbody>
</table>

Prerequisites for BFD

BFD has the following prerequisites:

- You must enable the BFD feature.
- Disable Internet Control Message Protocol (ICMP) redirect messages on BFD-enabled interfaces.
- Disable the IP packet verification check for identical IP source and destination addresses.
- See other detailed prerequisites that are listed with the configuration tasks.

Guidelines and Limitations

BFD has the following configuration guidelines and limitations:

- The QSFP 40/100-G BiDi comes up in the highest possible speed available on the port. For example, in the Cisco Nexus 93180LC-EX switch it comes up as 40 G in the first 28 ports and 100 G in the last 4 ports. If you need to connect to 40-G SR4 BiDi, the speed on the 40/100-G BiDi needs to be set to 40 G.

- Forming BFD neighbours on a vPC VLAN through an orphan port is not supported on Cisco Nexus 9000 Switches.

- Beginning with Cisco NX-OS Release 9.2(1), QSFP-40/100-SRBD comes up in the speed of 100-G and interoperate with other QSFP-40/100-SRBD at either 40-G or 100-G speed on Cisco Nexus 9500 Switches with the N9K-X9636C-RX line card. The QSFP-40/100-SRBD can also interoperate with QSFP-40G-SR-BD at 40G speeds. However to operate at 40G speed, you must configure the speed as 40G.

- The dual rate BiDi does not support auto speed sensing. With speed auto, the QSFP always defaults to 100 G rather than the speed of the link partner.

This feature is supported on the following Cisco Nexus switches:
• show commands with the `internal` keyword are not supported.

• Cisco Nexus 9000 Series switches supports BFD per-member link.

• BFD per-member link support is added on Cisco Nexus 9000 Series switches.

• BFD supports BFD version 1.

• BFD supports IPv4 and IPv6.

• BFD supports OSPFv3.

• BFD supports IS-ISv6.

• BFD supports BGPv6.

• BFD supports EIGRPv6.

• BFD supports only one session per address family, per interface.

• BFD supports single-hop BFD.

• BFD for BGP supports single-hop EBGP and iBGP peers.

• BFD supports keyed SHA-1 authentication.

• BFD supports the following Layer 3 interfaces—physical interfaces, port channels, subinterfaces, and VLAN interfaces.

• BFD depends on a Layer 3 adjacency information to discover topology changes, including Layer 2 topology changes. A BFD session on a VLAN interface (SVI) may not be up after the convergence of the Layer 2 topology if there is no Layer 3 adjacency information available.

• For BFD on a static route between two devices, both devices must support BFD. If one or both of the devices do not support BFD, the static routes are not programmed in the Routing Information Base (RIB).

• Port channel configuration limitations:
  • For Layer 3 port channels used by BFD, you must enable LACP on the port channel.
  • For Layer 2 port channels used by SVI sessions, you must enable LACP on the port channel.

• SVI limitations:
  • An ASIC reset causes traffic disruption for other ports and it can cause the SVI sessions on the other ports to flap. For example, if the carrier interface is a virtual port channel (vPC), BFD is not supported over the SVI interface and it could cause a trigger for an ASIC reset. When a BFD session is over
SVI using virtual port channel (vPC) peer-link, the BFD echo function is not supported. You must disable the BFD echo function for all sessions over SVI between vPC peer nodes.

An SVI on the Cisco Nexus series switches should not be configured to establish a BFD neighbor adjacency with a device connected to it via a vPC. This is because the BFD keepalives from the neighbour, if sent over the vPC member link connected to the vPC peer-switch, do not reach this SVI causing the BFD adjacency to fail.

- When you change the topology (for example, add or delete a link into a VLAN, delete a member from a Layer 2 port channel, and so on), the SVI session could be affected. It may go down first and then come up after the topology discovery is finished.

- BFD over FEX HIF interfaces is not supported.

- When a BFD session is over SVI using virtual port-channel (vPC) peer-link (either BCM or GEM based ports), the BFD echo function is not supported. You must disable the BFD echo function for all sessions over SVI between vPC peer nodes using the `no bfd echo` command at the SVI configuration level.

Tip

If you do not want the SVI sessions to flap and you need to change the topology, you can disable the BFD feature before making the changes and reenable BFD after the changes have been made. You can also configure the BFD timer to be a large value (for example, 5 seconds), and change it back to a fast timer after the above events complete.

- When you configure the BFD Echo function on the distributed Layer 3 port channels, reloading a member module flaps the BFD session hosted on that module, which results in a packet loss.

If you connect the BFD peers directly without a Layer 2 switch in between, you can use the BFD per-link mode as an alternative solution.

Note

Using BFD per-link mode and subinterface optimization simultaneously on a Layer 3 port channel is not supported.

- When you specify a BFD neighbor prefix in the `clear {ip | ipv6} route prefix` command, the BFD echo session will flap.

- The `clear {ip | ipv6} route *` command causes BFD echo sessions to flap.

- HSRP for IPv4 is supported with BFD.

- BFD packets generated by the Cisco NX-OS device linecards are sent with COS 6/DSCP CS6. The DSCP/COS values for BFD packets are not user configurable.

- When configuring BFDv6 in no-bfd-echo mode, it is recommended to run with timers of 150 ms with a multiplier of 3.

- BFDv6 is not supported for VRRPv3 and HSRP for v6.

- IPv6 `eigrp bfd` cannot be disabled on an interface.

• Port channel configuration notes:
  • When the BFD per-link mode is configured, the BFD echo function is not supported. You must disable the BFD echo function using the `no bfd echo` command before configuring the `bfd per-link` command.
  • Configuring BFD per-link with link-local is not supported.
  • The supported platforms include Cisco Nexus 9500 Series switches with N9K-X9636C-R, N9K-X9636Q-R, N9K-X9636C-RX line cards.

Default Settings

The following table lists the default settings for BFD parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>BFD feature</td>
<td>Disabled</td>
</tr>
<tr>
<td>Required minimum receive interval</td>
<td>50 milliseconds</td>
</tr>
<tr>
<td>Desired minimum transmit interval</td>
<td>50 milliseconds</td>
</tr>
<tr>
<td>Detect multiplier</td>
<td>3</td>
</tr>
<tr>
<td>Echo function</td>
<td>Enabled</td>
</tr>
<tr>
<td>Mode</td>
<td>Asynchronous</td>
</tr>
<tr>
<td>Port-channel</td>
<td>Logical mode (one session per source-destination pair address)</td>
</tr>
<tr>
<td>Slow timer</td>
<td>2000 milliseconds</td>
</tr>
<tr>
<td>Startup timer</td>
<td>5 seconds</td>
</tr>
</tbody>
</table>

Configuring BFD

Configuration Hierarchy

You can configure BFD at the global level and at the interface level. The interface configuration overrides the global configuration.

For physical ports that are members of a port channel, the member port inherits the master port channel BFD configuration.
Task Flow for Configuring BFD

Follow these steps in the following sections to configure BFD:

- Enabling the BFD Feature.
- Configuring Global BFD Parameters or Configuring BFD on an Interface.

Enabling the BFD Feature

You must enable the BFD feature before you can configure BFD on an interface and protocol.

Note

Use the `no feature bfd` command to disable the BFD feature and remove all associated configuration.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>no feature bfd</td>
<td>Disables the BFD feature and removes all associated configuration.</td>
</tr>
</tbody>
</table>

Example:

```
switch(config)# no feature bfd
```

SUMMARY STEPS

1. `configure terminal`
2. `feature bfd`
3. `show feature | include bfd`
4. `copy running-config startup-config`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td></td>
</tr>
<tr>
<td>configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# configure terminal</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2</th>
<th>feature bfd</th>
<th>Enables the BFD feature.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>switch(config)# feature bfd</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>show feature</th>
<th>include bfd</th>
<th>(Optional) Displays enabled and disabled features.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>switch(config)# show feature</td>
<td>include bfd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>copy running-config startup-config</td>
<td>(Optional) Saves the configuration change.</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------</td>
<td>------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>switch(config)# copy running-config startup-config</td>
<td></td>
</tr>
</tbody>
</table>

### Configuring Global BFD Parameters

You can configure the BFD session parameters for all BFD sessions on the device. The BFD session parameters are negotiated between the BFD peers in a three-way handshake.

See the Configuring BFD on an Interface section to override these global session parameters on an interface.

**Before you begin**

Enable the BFD feature.

**SUMMARY STEPS**

1. configure terminal
2. bfd interval mintx min_rx msec multiplier value
3. bfd slow-timer [interval]
4. [no] bfd startup-timer [seconds]
5. bfd echo-interface loopback interface number
6. show running-config bfd
7. copy running-config startup-config

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step 1</th>
<th>configure terminal</th>
<th>Enters configuration mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td></td>
<td>switch# configure terminal switch(config)#</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2</th>
<th>bfd interval mintx min_rx msec multiplier value</th>
<th>Configures the BFD session parameters for all BFD sessions on the device. This command overrides these values by configuring the BFD session parameters on an interface. The mintx and msec range is from 50 to 999 milliseconds and the default is 50. The multiplier range is from 1 to 50. The multiplier default is 3.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td></td>
<td>switch(config)# bfd interval 50 min_rx 50 multiplier 3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>bfd slow-timer [interval]</th>
<th>Configures the slow timer used in the echo function. This value determines how fast BFD starts up a new session and at what speed the asynchronous sessions use for BFD control packets when the echo function is enabled. The slow-timer value is used as the new control packet interval, while the echo packets use the configured BFD intervals.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td></td>
<td>switch(config)# bfd slow-timer 2000</td>
</tr>
</tbody>
</table>
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>The echo packets are used for link failure detection, while the control packets at the slower rate maintain the BFD session. The range is from 1000 to 3000 milliseconds. The default is 2000.</td>
<td></td>
</tr>
</tbody>
</table>

### Step 4

<table>
<thead>
<tr>
<th>[no] bfd startup-timer [seconds]</th>
</tr>
</thead>
</table>

**Example:**

```
switch(config)# bfd startup-timer 20
```

Configures the BFD startup timer, which delays the startup time for BFD sessions in order to give the routes that are being used by local and remote routers time to settle down in the hardware. Using this feature can prevent BFD flaps in higher scale scenarios. The range is from 0 to 30 seconds. The default is 5 seconds.

The `bfd startup-timer 0` command disables the BFD startup timer.

The `no bfd startup-timer` command sets the BFD startup timer to 5 seconds (the default value).

### Step 5

<table>
<thead>
<tr>
<th>bfd echo-interface loopback interface number</th>
</tr>
</thead>
</table>

**Example:**

```
switch(config)# bfd echo-interface loopback 1 3
```

Configures the interface used for Bidirectional Forwarding Detection (BFD) echo frames. This command changes the source address for the echo packets to the one configured on the specified loopback interface. The interface number range is from 0 to 1023.

### Step 6

<table>
<thead>
<tr>
<th>show running-config bfd</th>
</tr>
</thead>
</table>

**Example:**

```
switch(config)# show running-config bfd
```

(Optional) Displays the BFD running configuration.

### Step 7

<table>
<thead>
<tr>
<th>copy running-config startup-config</th>
</tr>
</thead>
</table>

**Example:**

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

(Optional) Saves the configuration change.

### Configuring BFD on an Interface

You can configure the BFD session parameters for all BFD sessions on an interface. The BFD session parameters are negotiated between the BFD peers in a three-way handshake.

This configuration overrides the global session parameters for the configured interface.

**Before you begin**

Ensure that Internet Control Message Protocol (ICMP) redirect messages are disabled on BFD-enabled interfaces. Use the `no ip redirects` command or the `no ipv6 redirects` command on the interface.

Enable the BFD feature. See the Enabling the BFD Feature section.

**SUMMARY STEPS**

1. `configure terminal`
2. `interface int-if`
3. `bfd interval mintx min_rx msec multiplier value`
4. `bfd authentication keyed-sha1 keyid id key ascii_key`
5. `show running-config bfd`
6. `copy running-config startup-config`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure terminal</code></td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch# <code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>interface int-if</code></td>
<td>Enters interface configuration mode. Use the ? keyword to display the supported interfaces.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)# <code>interface ethernet 2/1</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>bfd interval mintx min_rx msec multiplier value</code></td>
<td>Configures the BFD session parameters for all BFD sessions on the device. This command overrides these values by configuring the BFD session parameters on an interface. The <code>mintx</code> and <code>msec</code> range is from 50 to 999 milliseconds and the default is 50. The multiplier range is from 1 to 50. The multiplier default is 3.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# <code>bfd interval 50 min_rx 50 multiplier 3</code></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>bfd authentication keyed-sha1 keyid id key ascii_key</code></td>
<td>(Optional) Configures SHA-1 authentication for all BFD sessions on the interface. The <code>ascii_key</code> string is a secret key shared among BFD peers. The <code>id</code> value, a number between 0 and 255, is assigned to this particular <code>ascii_key</code>. BFD packets specify the key by <code>id</code>, allowing the use of multiple active keys. To disable SHA-1 authentication on the interface, use the <code>no</code> form of the command.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# <code>bfd authentication keyed-sha1 keyid 1 ascii_key cisco123</code></td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td><code>show running-config bfd</code></td>
<td>(Optional) Displays the BFD running configuration.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)# <code>show running-config bfd</code></td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td><code>copy running-config startup-config</code></td>
<td>(Optional) Saves the configuration change.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)# <code>copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>

### Configuring BFD on a Port Channel

You can configure the BFD session parameters for all BFD sessions on a port channel. If per-link mode is used for Layer 3 port channels, BFD creates a session for each link in the port channel and provides an aggregate result to client protocols. For example, if the BFD session for one link on a port channel is up, BFD
informs client protocols, such as OSPF, that the port channel is up. The BFD session parameters are negotiated between the BFD peers in a three-way handshake.

This configuration overrides the global session parameters for the configured port channel. The member ports of the port channel inherit the port channel BFD session parameters.

**Before you begin**

Ensure that you enable LACP on the port channel before you enable BFD.

Ensure that Internet Control Message Protocol (ICMP) redirect messages are disabled on BFD-enabled interfaces. Use the `no ip redirects` command on the interface.

Enable the BFD feature. See the Enabling the BFD Feature section.

### SUMMARY STEPS

1. `configure terminal`
2. `interface port-channel number`
3. `bfd per-link`
4. `bfd interval minRx minRx msec multiplier value`
5. `bfd authentication keyed-sha1 keyid id key ascii_key`
6. `show running-config bfd`
7. `copy running-config startup-config`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> switch# configure terminal switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface port-channel number</td>
<td>Enters port-channel configuration mode. Use the ? keyword to display the supported number range.</td>
</tr>
<tr>
<td><strong>Example:</strong> switch(config)# interface port-channel 2 switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> bfd per-link</td>
<td>Configures the BFD sessions for each link in the port channel.</td>
</tr>
<tr>
<td><strong>Example:</strong> switch(config-if)# bfd per-link</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> bfd interval minRx minRx msec multiplier value</td>
<td>(Optional) Configures the BFD session parameters for all BFD sessions on the port channel. This command overrides these values by configuring the BFD session parameters. The <code>minRx</code> and <code>msec</code> range is from 50 to 999 milliseconds and the default is 50. The multiplier range is from 1 to 50. The multiplier default is 3.</td>
</tr>
<tr>
<td><strong>Example:</strong> switch(config-if)# bfd interval 50 minRx 50 msec multiplier 3</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| **Step 5** | bfd authentication keyed-sha1 keyid id key ascii_key  
Example:  
switch(config-if)# bfd authentication keyed-sha1 keyid 1 ascii_key cisco123 |
| (Optional) Configures SHA-1 authentication for all BFD sessions on the interface. The ascii_key string is a secret key shared among BFD peers. The id value, a number between 0 and 255, is assigned to this particular ascii_key. BFD packets specify the key by id, allowing the use of multiple active keys. To disable SHA-1 authentication on the interface, use the no form of the command. |
| **Step 6** | show running-config bfd  
Example:  
switch(config-if)# show running-config bfd |
| (Optional) Displays the BFD running configuration. |
| **Step 7** | copy running-config startup-config  
Example:  
switch(config-if)# copy running-config startup-config |
| (Optional) Saves the configuration change. |

### Configuring the BFD Echo Function

You can configure the BFD echo function on one or both ends of a BFD-monitored link. The echo function slows down the required minimum receive interval, based on the configured slow timer. The RequiredMinEchoRx BFD session parameter is set to zero if the echo function is disabled. The slow timer becomes the required minimum receive interval if the echo function is enabled.

**Before you begin**

Enable the BFD feature. See the Enabling the BFD Feature section.

Configure the BFD session parameters. See the Configuring Global BFD Parameters section on or the Configuring BFD on an Interface section.

Ensure that Internet Control Message Protocol (ICMP) redirect messages are disabled on BFD-enabled interfaces. Use the **no ip redirects** command on the interface.

Ensure that the IP packet verification check for identical IP source and destination addresses is disabled. Use the **no hardware ip verify address identical** command. See the Cisco Nexus 9000 Series NX-OS Unicasting Routing Configuration Guide for more information about this command.

**SUMMARY STEPS**

1. configure terminal  
2. bfd slow-timer echo-interval  
3. interface int-if  
4. bfd echo  
5. show running-config bfd  
6. copy running-config startup-config
### Detailed Steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure terminal</code></td>
<td>Enters configuration mode.</td>
</tr>
</tbody>
</table>
|       | `switch# configure terminal`  
|       | `switch(config)#` | |
| Step 2 | `bfd slow-timer echo-interval` | Configures the slow timer used in the echo function. This value determines how fast BFD starts up a new session and is used to slow down the asynchronous sessions when the BFD echo function is enabled. This value overwrites the required minimum receive interval when the echo function is enabled. The range is from 1000 to 30000 milliseconds. The default is 2000. |
|       | `switch(config)# bfd slow-timer 2000` | |
| Step 3 | `interface int-if` | Enters interface configuration mode. Use the ? keyword to display the supported interfaces. |
|       | `switch(config)# interface ethernet 2/1`  
|       | `switch(config-if)#` | |
| Step 4 | `bfd echo` | Enables the echo function. The default is enabled. |
|       | `switch(config-if)# bfd echo` | |
| Step 5 | `show running-config bfd` | (Optional) Displays the BFD running configuration. |
|       | `switch(config-if)# show running-config bfd` | |
| Step 6 | `copy running-config startup-config` | (Optional) Saves the configuration change. |
|       | `switch(config-if)# copy running-config startup-config` | |

### Configuring Per-Member Link BFD Sessions

BFD per-member link support is added on Cisco Nexus 9000 Series switches. See the following sections for more information.

### BFD Enhancement to Address Per-link Efficiency

The Bidirectional Forwarding (BFD) enhancement to address per-link efficiency, called as IETF Micro BFD, lets you configure the individual BFD sessions on every Link Aggregation Group (LAG) member interfaces (as defined in RFC 7130).

With this enhancement, the BFD sessions run on each member link of the port-channel. If BFD detects a link failure, the member link is removed from the forwarding table. This mechanism delivers faster failure detection as the BFD sessions are created on an individual port-channel interface.
The BFD sessions running on member links of the port-channel are called as Micro BFD sessions. You can configure RFC 7130 BFD over main port-channel interface, that performs bandwidth monitoring over LAG by having one Micro BFD session over each member. If any of the member port goes down, the port is removed from the forwarding table and this prevents traffic disruption on that member.

Micro BFD sessions are supported for both LACP and non-LACP based-port channels. For more information on how to configure Micro BFD sessions, see Configuring Micro BFD Sessions.

Limitations of the IETF Bidirectional Forwarding Detection

See the following limitations of the IETF Bidirectional Forwarding Detection:

- **BFD Limitations**
  - It cannot co-exist with BFD over logical port-channels or proprietary BFD per-member links. BFD IPv6 logical/proprietary per-link session is also not supported when BFD IETF IPv4 is configured on PC.
  - IETF BFD IPv6 is not supported.
  - Echo functionality is not supported for Micro-BFD sessions.
  - Port-channel interfaces should be directly connected between two switches that are running the BFD sessions. No intermediate Layer 2 switches are expected.

- **EthPCM/LACP Limitations**
  - If a LACP port-channel has members in hot-standby state, BFD failure in one of the active links may not cause the hot-standby link to come up directly. Once the active link with BFD failure goes down, the hot-standby member becomes active. However, it may not be able to prevent the port-channel from going down before the hot-standby link comes up, in cases where port-channel min-link condition is hit.

- **General Limitations:**
  - It is supported only on Layer 3 port-channels.
  - It is not supported on the following:
    - vPC
    - Layer 3 sub-interfaces
    - Layer 2 port-channels/Layer 2 Fabric Path
    - FPC/HIF PC
    - Layer 3 sub-interfaces
    - VSI over port-channels

Guidelines for Migration/Configuration of IETF Per-Member Sessions:

See the following guidelines for migration/configuration of IETF per-member sessions:
• The logical BFD sessions that are created using the routing protocols over port-channel sub-interfaces (where RFC 7130 cannot run) are still supported. The main port-channel interface however does not support both logical and RFC 7130 sessions that co-exist. It can support only either of them.

• You can configure RFC 7130 BFD over the main port-channel interface that perform bandwidth monitoring over the LAG by having one Micro-BFD session over each member. If any of the member port goes down, BFD notifies it to the port-channel manager that removes the port from the LTL, thereby preventing blackholing of the traffic on that member.

• If the minimum number of links required to have the port-channel operationally up is not met in the above case, the port-channel is brought down by the port-channel manager. This in turn brings down the port-channel sub-interfaces if they are configured and thereby the logical BFD session also comes down notifying the routing protocol.

• When you are using RFC 7130 on the main port-channel and logical BFD on the sub-interfaces, the logical BFD session should be run with lesser aggressive timers than the RFC 7130 BFD session. You can have RFC 7130 configured on the port-channel interface or you can have it configured in conjunction with the logical BFD sessions on the port-channel sub-interfaces.

• When a proprietary per-link is configured, enabling IETF Micro-BFD sessions is not allowed on a port channel and vice-versa. You have to remove the proprietary per-link configuration. Current implementation of proprietary per-link does not allow changing the configuration (no per-link), if there is any BFD session that is bootstrapped by the applications. You need to remove the BFD tracking on the respective applications and remove per-link configuration. The migration path from the proprietary per-link to IETF Micro-BFD is as follows:
  • Remove the BFD configuration on the applications.
  • Remove the per-link configuration.
  • Enable the IETF Micro-BFD command.
  • Enable BFD on the applications.

The same migration path can be followed for proprietary BFD to IETF Micro-BFD on the main port-channel interface.

**Configuring Port Channel Interface**

**Before you begin**

Ensure that the BFD feature is enabled.

**SUMMARY STEPS**

1. switch(config)# interface port-channel port-number
2. switch(config)# no switchport

**DETAILED STEPS**

**Step 1**

switch(config)# interface port-channel port-number

Configures interface port-channel.
Step 2

```
switch(config)# no switchport
```
Configures interface as Layer 3 port-channel.

---

**What to do next**

- Configuring BFD Start Timer
- Enabling IETF Per-link BFD

### Configuring BFD Start Timer

Complete the following steps to configure the BFD start timer:

**SUMMARY STEPS**

1. `switch(config)# interface port-channel port-number`

**DETAILED STEPS**

```
switch(config)# interface port-channel port-number
```
Configures the BFD start timer for a port-channel.

**Note**

The default value is infinite (that is no timer is running). For start timer to work, configure start timer value before completing the port-channel BFD configurations (for example, before the port-channel bfd track-member-link and port-channel bfd destination are configured for Layer 3 port-channel interface with the active members).

---

**What to do next**

- Enabling IETF Per-link BFD
- Configuring BFD Destination IP Address

### Enabling IETF Per-link BFD

**SUMMARY STEPS**

1. `switch(config-if)# port-channel bfd track-member-link`

**DETAILED STEPS**

```
switch(config-if)# port-channel bfd track-member-link
```

---

Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide, Release 9.2(x)
Enables IETF BFD on port-channel interface.

What to do next
- Configuring BFD Destination IP Address
- Verifying Micro BFD Session Configurations

Configuring BFD Destination IP Address
Complete the following steps to configure the BFD destination IP address:

SUMMARY STEPS
1. `switch(config-if)# port-channel bfd destination ip-address`

DETAILED STEPS

switch(config-if)# port-channel bfd destination ip-address
Configures an IPv4 address to be used for the BFD sessions on the member links.

What to do next
- Verifying Micro BFD Sessions Configuration

Verifying Micro BFD Session Configurations
Use the following commands to verify the Micro BFD session configurations.

SUMMARY STEPS
1. Displays the port-channel and port-channel member operational state.
2. `switch# show bfd neighbors`
3. `switch# show bfd neighbors details`
4. `switch# show tech-support bfd`
5. `switch# show tech-support lacp all`
6. `switch# show running-config interface port-channel port-channel-number`

DETAILED STEPS

Step 1 Displays the port-channel and port-channel member operational state.

`switch# show port-channel summary`

Step 2 `switch# show bfd neighbors`
Displays Micro BFD sessions on port-channel members.

**Step 3**
```
switch# show bfd neighbors details
```
Displays BFD session for a port channel interface and the associated Micro BFD sessions on members.

**Step 4**
```
switch# show tech-support bfd
```
Displays the technical support information for BFD.

**Step 5**
```
switch# show tech-support lacp all
```
Displays the technical support information for Ethernet Port Manager, Ethernet Port-channel Manager, and LACP.

**Step 6**
```
switch# show running-config interface port-channel port-channel-number
```
Displays the running configuration information of the port-channel interface.

---

**Examples: Configuring Micro BFD Sessions**

See the following examples for configuring Micro BFD sessions.

**Configuring Micro BFD Sessions**

In this example, the following topology is used.

*Figure 8: Configuring Micro BFD Session*

The sample configuration of switch 1 is as follows:

```
feature bfd
configure terminal
   interface port-channel 10
      port-channel bfd track-member-link
      port-channel bfd destination 10.1.1.2
      port-channel bfd start 60
      ip address 10.1.1.1/24
```

The sample configuration of switch 2 is as follows:

```
feature bfd
configure terminal
   interface port-channel 10
      port-channel bfd track-member-link
      port-channel bfd destination 10.1.1.1
```
Verifying Micro BFD Sessions Configuration

The following example displays the show output of the `show running-config interface port-channel <port-channel>` command, `show port-channel summary`, `show bfd neighbors vrf internet_routes`, and `show bfd neighbors interface port-channel <port-channel> vrf internet_routes details` commands.

```
switch# show running-config interface port-channel 1001
!Command: show running-config interface port-channel1001
!Time: Fri Oct 21 09:08:00 2016
version 7.0(3)I5(1)
interface port-channel1001
  no switchport
  vrf member internet_routes
  port-channel bfd track-member-link
  port-channel bfd destination 40.4.1.2
  ip address 40.4.1.1/24
  ipv6 address 2001:40:4:1::1/64

switch# show port-channel port-profile

switch# show port-channel summary
Flags: D - Down P - Up in port-channel (members)
I - Individual H - Hot-standby (LACP only)
s - Suspended r - Module-removed
b - BFD Session Wait S - Switched R - Routed
U - Up (port-channel)
p - Up in delay-lacp mode (member)
M - Not in use. Min-links not met
--------------------------------------------------------------------------------
Group Port- Type Protocol Member Ports
Channel
--------------------------------------------------------------------------------
1001 Po1001(RU) Eth LACP Eth1/11/1(P) Eth1/11/2(P) Eth1/12/1(P)
  Eth1/12/2(P)

switch# show bfd neighbors vrf internet_routes
OurAddr NeighAddr LD/RD RH/RS Holdown(mult)
State Int Vrf
40.4.1.1 40.4.1.2 1090519041/0 Up N/A(3) Up
internet_routes
40.4.1.1 40.4.1.2 1090519042/1090519051 Up 819(3) Up
internet_routes
40.4.1.1 40.4.1.2 1090519043/1090519052 Up 819(3) Up
internet_routes
40.4.1.1 40.4.1.2 1090519044/1090519053 Up 819(3) Up
internet_routes
40.4.1.1 40.4.1.2 1090519045/1090519054 Up 819(3) Up
internet_routes

switch#
```

```
switch# show bfd neighbors interface port-channel 1001 vrf internet_routes details
OurAddr NeighAddr LD/RD RH/RS Holdown(mult)
State Int Vrf
```

```
Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide, Release 9.2(x)

port-channel bfd start 60
  ip address 10.1.1.2/24

Examples: Configuring Micro BFD Sessions

```
### Session state is Up

Local Diag: 0
Registered protocols: `eth_port_channel`
Uptime: 1 days 11 hrs 4 mins 8 secs
Hosting LC: 0, Down reason: None, Reason not-hosted: None
Parent session, please check port channel config for member info

### switch#

```bash
switch# show bfd neighbors interface ethernet 1/12/1 vrf internet_routes details
```

<table>
<thead>
<tr>
<th>OurAddr</th>
<th>NeighAddr</th>
<th>LD/RD</th>
<th>RH/RS</th>
<th>Holdown(mult)</th>
</tr>
</thead>
<tbody>
<tr>
<td>40.4.1.1</td>
<td>40.4.1.2</td>
<td>1090519042/1090519051 Up</td>
<td>604(3) Up</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>Int</td>
<td>Vrf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eth1/12/1</td>
<td>internet_routes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Session state is Up and not using echo function
Local Diag: 0, Demand mode: 0, Poll bit: 0, Authentication: None
MinTxInt: 100000 us, MinRxInt: 100000 us, Multiplier: 3
Received MinRxInt: 300000 us, Received Multiplier: 3
Holdown (hits): 300 ms (0), Hello (hits): 300 ms (458317)
Rx Count: 427188, Rx Interval (ms) min/max/avg: 19/1668/295 last: 295 ms ago
Tx Count: 458317, Tx Interval (ms) min/max/avg: 275/275/275 last: 64 ms ago
Registered protocols: `eth_port_channel`
Uptime: 1 days 11 hrs 4 mins 24 secs
Last packet: Version: 1 - Diagnostic: 0
State bit: Up - Demand bit: 0
Poll bit: 0 - Final bit: 0
Multiplier: 3 - Length: 24
My Discr.: 1090519051 - Your Discr.: 1090519042
Min tx interval: 300000 - Min rx interval: 300000
Min Echo interval: 300000 - Authentication bit: 0
Hosting LC: 1, Down reason: None, Reason not-hosted: None
Member session under parent interface Po1001

### switch#

```bash
switch# show bfd neighbors interface ethernet 1/12/2 vrf internet_routes details
```

<table>
<thead>
<tr>
<th>OurAddr</th>
<th>NeighAddr</th>
<th>LD/RD</th>
<th>RH/RS</th>
<th>Holdown(mult)</th>
</tr>
</thead>
<tbody>
<tr>
<td>40.4.1.1</td>
<td>40.4.1.2</td>
<td>1090519043/1090519052 Up</td>
<td>799(3) Up</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>Int</td>
<td>Vrf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eth1/12/2</td>
<td>internet_routes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Session state is Up and not using echo function
Local Diag: 0, Demand mode: 0, Poll bit: 0, Authentication: None
MinTxInt: 100000 us, MinRxInt: 100000 us, Multiplier: 3
Received MinRxInt: 300000 us, Received Multiplier: 3
Holdown (hits): 300 ms (0), Hello (hits): 300 ms (458336)
Rx Count: 427207, Rx Interval (ms) min/max/avg: 19/1668/295 last: 100 ms ago
Tx Count: 458336, Tx Interval (ms) min/max/avg: 275/275/275 last: 251 ms ago
Registered protocols: `eth_port_channel`
Uptime: 1 days 11 hrs 4 mins 30 secs
Last packet: Version: 1 - Diagnostic: 0
State bit: Up - Demand bit: 0
Poll bit: 0 - Final bit: 0
Multiplier: 3 - Length: 24
My Discr.: 1090519052 - Your Discr.: 1090519043
Min tx interval: 300000 - Min rx interval: 300000
Min Echo interval: 300000 - Authentication bit: 0
Hosting LC: 1, Down reason: None, Reason not-hosted: None
Member session under parent interface Po1001

switch#
Configuring BFD Support for Routing Protocols

Configuring BFD on BGP

You can configure BFD for the Border Gateway Protocol (BGP).

**Before you begin**

Enable the BFD feature. See the Enabling the BFD Feature section.

Configure the BFD session parameters. See the Configuring Global BFD Parameters section or the Configuring BFD on an Interface section.

Enable the BGP feature. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information.

**SUMMARY STEPS**

1. `configure terminal`
2. `router bgp as-number`
3. `neighbor (ip-address | ipv6-address) remote-as as-number`
4. `bfd`
5. `update-source interface`
6. `show running-config bgp`
7. `copy running-config startup-config`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>Example: <code>switch# configure terminal</code> <code>switch(config)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> router bgp as-number</td>
<td>Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.</td>
</tr>
<tr>
<td>Example: <code>switch(config)# router bgp 64496</code> <code>switch(config-router)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> neighbor (ip-address</td>
<td>ipv6-address) remote-as as-number</td>
</tr>
<tr>
<td>Example: <code>switch(config-router)# neighbor 209.165.201.1 remote-as 64497</code> <code>switch(config-router-neighbor)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> bfd</td>
<td>Enables BFD for this BGP peer.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
Configuring BFD on EIGRP

You can configure BFD for the Enhanced Interior Gateway Routing Protocol (EIGRP).

Before you begin

Enable the BFD feature. See the Enabling the BFD Feature section.

Configure the BFD session parameters. See the Configuring Global BFD Parameters section or the Configuring BFD on an Interface section.

Enable the EIGRP feature. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information.

SUMMARY STEPS

1. configure terminal
2. router eigrp instance-tag
3. bfd [ipv4 | ipv6]
4. interface int-if
5. ip eigrp instance-tag bfd
6. show ip eigrp [vrf vrf-name] [ interfaces if]
7. copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
</tbody>
</table>

允许 BGP 会话使用特定接口的主 IP 地址作为本地地址来建立更新源。

步骤5

```
switch(config-router-neighbor)# bfd
```

允许 BGP 会话使用特定接口的主 IP 地址作为本地地址来建立更新源。

步骤6

```
switch(config-router-neighbor)# update-source interface
```

允许 BGP 会话使用特定接口的主 IP 地址作为本地地址来建立更新源。

步骤7

```
switch(config-router-neighbor)# show running-config bgp
```

显示 BGP 运行配置。

步骤8

```
switch(config-router-neighbor)# copy running-config startup-config
```

保存配置更改。

配置 EIGRP 上的 BFD

您可以为增强的内部路由协议 (EIGRP) 配置 BFD。
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| `switch# configure terminal`  
`switch(config)#`                                                                                                                                                                         |                                                                                                                                                                                                         |
| **Step 2**  
**router eigrp instance-tag**  
**Example:**  
`s witch(config)# router eigrp Test1  
switch(config-router)#`                                                                                                                                                                                                                       | Creates a new EIGRP process with the configured instance tag. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.  
If you configure an instance-tag that does not qualify as an AS number, you must use the `autonomous-system` to configure the AS number explicitly or this EIGRP instance will remain in the shutdown state. |
| **Step 3**  
**bfd [ipv4 | ipv6]**  
**Example:**  
`s witch(config-router-neighbor)# bfd ipv4`                                                                                                                                                                                                                                                  | (Optional) Enables BFD for all EIGRP interfaces.                                                                                                                                                           |
| **Step 4**  
**interface int-if**  
**Example:**  
`s witch(config-router-neighbor)# interface ethernet 2/1  
switch(config-if)#`                                                                                                                                                                                                                | Enters interface configuration mode. Use the `?` keyword to display the supported interfaces.                                                                                                               |
| **Step 5**  
**ip eigrp instance-tag bfd**  
**Example:**  
`s witch(config-if)# ip eigrp Test1 bfd`                                                                                                                                                                                                                                                   | (Optional) Enables or disables BFD on an EIGRP interface. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.  
The default is disabled.                                                                                                                                                                          |
| **Step 6**  
**show ip eigrp [vrf vrf-name] [ interfaces if]**  
**Example:**  
`s witch(config-if)# show ip eigrp`                                                                                                                                                                                                                                                      | (Optional) Displays information about EIGRP. The `vrf-name` can be any case-sensitive, alphanumeric string up to 32 characters.                                                                          |
| **Step 7**  
**copy running-config startup-config**  
**Example:**  
`s witch(config-if)# copy running-config startup-config`                                                                                                                                                                                                                                 | (Optional) Saves the configuration change.                                                                                                                                                               |

### Configuring BFD on OSPF

You can configure BFD for the Open Shortest Path First.

**Before you begin**

Enable the BFD feature. See the Enabling the BFD Feature section.

Configure the BFD session parameters. See the Configuring Global BFD Parameters section or the Configuring BFD on an Interface section.

Enable the OSPF feature. See the *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide* for more information.
### SUMMARY STEPS

1. configure terminal
2. router ospf instance-tag
3. bfd [ipv4 | ipv6]
4. interface int-if
5. ip ospf bfd
6. show ip ospf [vrf vrf-name] [ interfaces if]
7. copy running-config startup-config

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>router ospf instance-tag</td>
<td>Creates a new OSPF instance with the configured instance tag. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)# router ospf 200</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-router)#</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>bfd [ipv4</td>
<td>ipv6]</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-router)# bfd</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>interface int-if</td>
<td>Enters interface configuration mode. Use the ? keyword to display the supported interfaces.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-router)# interface ethernet 2/1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>ip ospf bfd</td>
<td>(Optional) Enables or disables BFD on an OSPF interface. The default is disabled.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# ip ospf bfd</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>show ip ospf [vrf vrf-name] [ interfaces if]</td>
<td>(Optional) Displays information about OSPF. The vrf-name can be any case-sensitive, alphanumeric string up to 32 characters.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# show ip ospf</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>copy running-config startup-config</td>
<td>(Optional) Saves the configuration change.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# copy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>running-config startup-config</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Bidirectional Forwarding Detection

Example Configurations for BFD on OSPF
Example configuration where BFD is enabled under a non-default VRF (OSPFv3 neighbors in vrf3).

```plaintext
configure terminal
  router ospfv3 10
  vrf vrf3
  bfd
```

Configuring BFD on IS-IS

You can configure BFD for the Intermediate System-to-Intermediate System (IS-IS) protocol.

Before you begin

Enable the BFD feature. See the Enabling the BFD Feature section.

Configure the BFD session parameters. See the Configuring Global BFD Parameters section or the Configuring BFD on an Interface section.

Enable the IS-IS feature. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information.

SUMMARY STEPS

1. `configure terminal`
2. `router isis instance-tag`
3. `bfd [ipv4 | ipv6]`
4. `interface int-if`
5. `isis bfd`
6. `show isis [vrf vrf-name] [ interface if]`
7. `copy running-config startup-config`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>configure terminal</code></td>
</tr>
<tr>
<td>Example:</td>
<td><code>switch# configure terminal</code></td>
</tr>
<tr>
<td></td>
<td><code>switch(config)#</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>router isis instance-tag</code></td>
</tr>
<tr>
<td>Example:</td>
<td><code>switch(config)# router isis 100</code></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-router)# net 49.0001.1720.1600.1001.00</code></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-router)# address-family ipv6 unicast</code></td>
</tr>
</tbody>
</table>
### Configuring BFD on IS-IS

**Example Configurations for BFD on IS-IS**

Example configuration for IS-IS where BFD is enabled under IPv4 and an IPv6 address family.

```
configure terminal
router isis isis-1
  bfd
  address-family ipv6 unicast
    bfd
```

### Configuring BFD on HSRP

You can configure BFD for the Hot Standby Router Protocol (HSRP). The active and standby HSRP routers track each other through BFD. If BFD on the standby HSRP router detects that the active HSRP router is down, the standby HSRP router treats this event as an active timer expiry and takes over as the active HSRP router.

The `show hsrp detail` command shows this event as BFD@Act-down or BFD@Sby-down.

**Before you begin**

Enable the BFD feature. See the Enabling the BFD Feature section.
Configure the BFD session parameters. See the Configuring Global BFD Parameters section or the Configuring BFD on an Interface section.

Enable the HSRP feature. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information.

SUMMARY STEPS

1. configure terminal
2. hsrp bfd all-interfaces
3. interface int-if
4. hsrp bfd
5. show running-config hsrp
6. copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** | **configure terminal**  
**Example:**  
switch# configure terminal  
switch(config)# | Enters global configuration mode. |
| **Step 2** | **hsrp bfd all-interfaces**  
**Example:**  
switch# hsrp bfd all-interfaces | (Optional) Enables or disables BFD on all HSRP interfaces.  
The default is disabled. |
| **Step 3** | **interface int-if**  
**Example:**  
switch(config-router)# interface ethernet 2/1  
switch(config-if)# | Enters interface configuration mode. Use the ? keyword to display the supported interfaces. |
| **Step 4** | **hsrp bfd**  
**Example:**  
switch(config-if)# hsrp bfd | (Optional) Enables or disables BFD on an HSRP interface.  
The default is disabled. |
| **Step 5** | **show running-config hsrp**  
**Example:**  
switch(config-if)# show running-config hsrp | (Optional) Displays the HSRP running configuration. |
| **Step 6** | **copy running-config startup-config**  
**Example:**  
switch(config-if)# copy running-config startup-config | (Optional) Saves the configuration change. |
Configuring BFD on VRRP

You can configure BFD for the Virtual Router Redundancy Protocol (VRRP). The active and standby VRRP routers track each other through BFD. If BFD on the standby VRRP router detects that the active VRRP router is down, the standby VRRP router treats this event as an active timer expiry and takes over as the active VRRP router.

The `show vrrp detail` command shows this event as BFD@Act-down or BFD@Sby-down.

**Before you begin**

Enable the BFD feature. See the Enabling the BFD Feature section.

Configure the BFD session parameters. See the Configuring Global BFD Parameters section or the Configuring BFD on an Interface section.

Enable the VRRP feature. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information.

**SUMMARY STEPS**

1. `configure terminal`
2. `interface int-if`
3. `vrrp group-no`  
4. `vrrp bfd address`
5. `show running-config vrrp`
6. `copy running-config startup-config`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config)#</code></td>
<td></td>
</tr>
</tbody>
</table>

| Step 2 interface int-if    | Enters interface configuration mode. Use the ? keyword to display the supported interfaces. |
| Example:                   |                                            |
| `switch(config)# interface` |                                            |
| `ethernet 2/1`             |                                            |
| `switch(config-if)#`       |                                            |

| Step 3 vrrp group-no       | Specifies the VRRP group number.           |
| Example:                   |                                            |
| `switch(config-if)# vrrp 2` |                                            |

| Step 4 vrrp bfd address    | Enables or disables BFD on a VRRP interface. The default is disabled. |
| Example:                   |                                            |
| `switch(config-if)# vrrp bfd` |                                            |
### Configuring BFD on PIM

You can configure BFD for the Protocol Independent Multicast (PIM) protocol.

#### Before you begin

Enable the BFD feature. See the Enabling the BFD Feature section.

Enable the PIM feature. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information.

#### SUMMARY STEPS

1. configure terminal
2. ip pim bfd
3. interface int-if
4. ip pim bfd-instance [disable]
5. show running-config pim
6. copy running-config startup-config

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> ip pim bfd</td>
<td>Enables BFD for PIM.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# ip pim bfd</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> interface int-if</td>
<td>Enters interface configuration mode. Use the ? keyword to display the supported interfaces.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# interface ethernet 2/1</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)#</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring BFD on Static Routes

You can configure BFD for static routes on an interface. You can optionally configure BFD on a static route within a virtual routing and forwarding (VRF) instance.

#### Before you begin
Enable the BFD feature. See the Enabling the BFD Feature section.

#### SUMMARY STEPS

1. `configure terminal`
2. `vrf context vrf-name`
3. `ip route route interface {nh-address | nh-prefix}`
4. `ip route static bfd interface {nh-address | nh-prefix}`
5. `show ip route static [vrf vrf-name]`
6. `copy running-config startup-config`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>configure terminal</code></td>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: <code>switch# configure terminal</code> <code>switch(config)#</code></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>vrf context vrf-name</code></td>
<td></td>
<td>(Optional) Enters VRF configuration mode.</td>
</tr>
<tr>
<td>Example: <code>switch(config)# vrf context Red</code> <code>switch(config-vrf)#</code></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Disabling BFD on an Interface

You can selectively disable BFD on an interface for a routing protocol that has BFD enabled at the global or VRF level.

To disable BFD on an interface, use one of the following commands in interface configuration mode:

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip route route interface (\text{nh-address}</td>
<td>\text{nh-prefix})</td>
</tr>
<tr>
<td>Example: switch(config-vrf)# ip route 192.0.2.1 ethernet 2/1 192.0.2.4</td>
<td></td>
</tr>
<tr>
<td>ip route static bfd interface (\text{nh-address}</td>
<td>\text{nh-prefix})</td>
</tr>
<tr>
<td>Example: switch(config-vrf)# ip route static bfd ethernet 2/1 192.0.2.4</td>
<td></td>
</tr>
<tr>
<td>show ip route static [vrf vrf-name]</td>
<td>(Optional) Displays the static routes.</td>
</tr>
<tr>
<td>Example: switch(config-vrf)# show ip route static vrf Red</td>
<td></td>
</tr>
<tr>
<td>copy running-config startup-config</td>
<td>(Optional) Saves the configuration change.</td>
</tr>
<tr>
<td>Example: switch(config-vrf)# copy running-config startup-config</td>
<td></td>
</tr>
</tbody>
</table>

#### Disabling BFD on an Interface

Disabling BFD on an Interface

Example configuration where BFD is disabled per interface.
configure terminal
interface port-channel 10
no ip redirects
ip address 22.1.10.1/30
ipv6 address 22:1:10::1/120
no ipv6 redirects
ip router ospf 10 area 0.0.0.0
ip ospf bfd disable /*** disables IPv4 BFD session for OSPF
ospfv3 bfd disable /*** disables IPv6 BFD session for OSPFv3

Configuring BFD Interoperability

Configuring BFD Interoperability in Cisco NX-OS Devices in a Point-to-Point Link

SUMMARY STEPS

1. configure terminal
2. interface port-channel int-if
3. ip ospf bfd
4. no ip redirects
5. bfd interval mintx min_rx msec multiplier value
6. exit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>interface port-channel int-if</td>
<td>Enters interface configuration mode. Use the ? keyword to display the supported interfaces.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# interface ethernet 2/1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ip ospf bfd</td>
<td>Enables BFD on an OSPFv2 interface. The default is disabled. OSPF is used as an example. You can enable BFD of any of the supported protocols.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# ip ospf bfd</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>no ip redirects</td>
<td>Prevents the device from sending redirects.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# no ip redirects</td>
<td></td>
</tr>
</tbody>
</table>
**Configuring BFD Interoperability in Cisco NX-OS Devices in a Switch Virtual Interface**

**SUMMARY STEPS**

1. configure terminal
2. interface port-channel vlan vlan-id
3. bfd interval mintx min_rx msec multiplier value
4. no ip redirects
5. ip address ip-address/length
6. ip ospf bfd
7. exit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: switch# configure terminal switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface port-channel vlan vlan-id</td>
<td>Creates a dynamic Switch Virtual Interface (SVI).</td>
</tr>
<tr>
<td>Example: switch(config)# interface vlan 998 switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> bfd interval mintx min_rx msec multiplier value</td>
<td>Configures the BFD session parameters for all BFD sessions on the device. The \textit{mintx} and \textit{msec} range is from 50 to 999 milliseconds and the default is 50. The multiplier range is from 1 to 50. The multiplier default is 3.</td>
</tr>
<tr>
<td>Example: switch(config-if)# bfd interval 50 min_rx 50 multiplier 3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> no ip redirects</td>
<td>Prevents the device from sending redirects.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring BFD Interoperability in Cisco NX-OS Devices in Logical Mode

#### SUMMARY STEPS

1. configure terminal
2. interface port-channel type number.subinterface-id
3. bfd interval min_tx min_rx msec multiplier value
4. no ip redirects
5. ip ospf bfd
6. exit

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>interface port-channel type number.subinterface-id</td>
<td>Enters port channel configuration mode. Use the ? keyword to display the supported number range.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# interface port-channel 50.2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>bfd interval min_tx min_rx msec multiplier value</td>
<td>Configures the BFD session parameters for all BFD sessions on the port channel. The min_tx and msec range is from 50 to 999 milliseconds and the default is 50. The multiplier range is from 1 to 50. The multiplier default is 3.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# bfd interval 50 min_rx 50 multiplier 3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>no ip redirects</td>
<td>Prevents the device from sending redirects.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Verifying BFD Interoperability in a Cisco Nexus 9000 Series Device

The following example shows how to verify BFD interoperability in a Cisco Nexus 9000 Series device.

```plaintext
switch# show bfd neighbors details
OurAddr NeighAddr LD/RD RH/RS Holdown(mult) State Int Vrf
10.1.1.1 10.1.1.2 1140850707/2147418093 Up 6393(4) Up Vlan2121 default
Session state is Up and using echo function with 50 ms interval
Local Diag: 0, Demand mode: 0, Poll bit: 0, Authentication: None
MinTxInt: 50000 us, MinRxInt: 2000000 us, Multiplier: 3
Received MinRxInt: 2000000 us, Received Multiplier: 4
Holdown (hits): 8000 ms (0), Hello (hits): 2000 ms (108)
Rx Count: 92, Rx Interval (ms) min/max/avg: 347/1996/1776 last: 1606 ms ago
Tx Count: 108, Tx Interval (ms) min/max/avg: 1515/1515/1515 last: 1233 ms ago
Registered protocols: ospf
Uptime: 0 days 0 hrs 2 mins 44 secs
Last packet: Version: 1 - Diagnostic: 0
State bit: Up - Demand bit: 0
Poll bit: 0 - Final bit: 0
Multiplier: 4 - Length: 24
My Discr.: 2147418093 - Your Discr.: 1140850707
Min tx interval: 2000000 - Min rx interval: 2000000
Min Echo interval: 1000000 - Authentication bit: 0
Hosting LC: 10, Down reason: None, Reason not-hosted: None
```

```plaintext
switch# show bfd neighbors details
OurAddr NeighAddr LD/RD RH/RS Holdown(mult) State Int Vrf
10.0.2.1 10.0.2.2 1140850695/131083 Up 270(3) Up Po14.121 default
Session state is Up and not using echo function
Local Diag: 0, Demand mode: 0, Poll bit: 0, Authentication: None
MinTxInt: 50000 us, MinRxInt: 500000 us, Multiplier: 3
Received MinRxInt: 1000000 us, Received Multiplier: 3
Holdown (hits): 300 ms (0), Hello (hits): 100 ms (3136283)
Rx Count: 2669290, Rx Interval (ms) min/max/avg: 12/1999/93 last: 29 ms ago
Tx Count: 3136283, Tx Interval (ms) min/max/avg: 77/77/77 last: 76 ms ago
Registered protocols: ospf
Uptime: 2 days 21 hrs 41 mins 45 secs
Last packet: Version: 1 - Diagnostic: 0
State bit: Up - Demand bit: 0
```

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch(config-if)# no ip redirects</td>
<td>Enables BFD on an OSPFv2 interface. The default is disabled. OSPF is used as an example. You can enable BFD of any of the supported protocols.</td>
</tr>
</tbody>
</table>

### Step 5

**Example:**

```
switch(config-if)# ip ospf bfd
```

Exits interface configuration mode and returns to EXEC mode.

---

**Verifying BFD Interoperability in a Cisco Nexus 9000 Series Device**

The following example shows how to verify BFD interoperability in a Cisco Nexus 9000 Series device.
Verifying the BFD Configuration

To display BFD configuration information, perform one of the following:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>show running-config bfd</td>
<td>Displays the running BFD configuration.</td>
</tr>
<tr>
<td>show startup-config bfd</td>
<td>Displays the BFD configuration that will be applied on the next system startup.</td>
</tr>
</tbody>
</table>

Monitoring BFD

Use the following commands to display BFD:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>show bfd neighbors [application name] [details]</td>
<td>Displays information about BFD for a supported application, such as BGP or OSPFv2.</td>
</tr>
<tr>
<td>show bfd neighbors [interface int-if] [details]</td>
<td>Displays information about BGP sessions on an interface.</td>
</tr>
<tr>
<td>show bfd neighbors [dest-ip ip-address] [src-ip ip-address] [details]</td>
<td>Displays information about the specified BGP session on an interface.</td>
</tr>
<tr>
<td>show bfd neighbors [vrf vrf-name] [details]</td>
<td>Displays information about BFD for a VRF.</td>
</tr>
<tr>
<td>show bfd [ipv4</td>
<td>ipv6] [neighbors]</td>
</tr>
</tbody>
</table>

BFD Multihop

Beginning with Cisco NX-OS Release 9.2(1), BFD multihop is supported on Cisco Nexus 9000 switches in compliance with RFC5883. You can now configure IPv4 BFD sessions over multihop routes. BFD multihop session is setup between a unique source and destination address pair. A multihop BFD session is associated with the link between a source-destination rather than an interface as in single hop BFD sessions. BFD multihop is not supported on IPv6.
BFD Multihop Number of Hops

BFD multihop sets the TTL field to the maximum limit, and it does not check the value on reception. The BFD code has no impact on the number of hops a BFD multihop packet can traverse. However, in most of the systems, it limits the number of hops to 255.

Guidelines and Limitations for BFD Multihop

BFD multihop has the following configuration guidelines and limitations:

- BFD multihop is supported on Cisco Nexus 9200 and 9300-EX platform switches and Cisco Nexus 9500 platform switches with N9K-X9700-EX line cards.
- Multihop BFD is identified with UDP Destination port 4784.
- The default interval timers for multihop BFD is 250 ms with multiplier 3.
- The maximum number of multihop BFD sessions supported is 100.
- The existing BFD authentication support is extended for multihop sessions.
- Echo mode is not supported for multihop BFD.
- Multihop with segment routing underlay is not supported.
- BFD multihop is not supported for BGP IPv6 multihop neighbors.

Configuring BFD Multihop Session Global Interval Parameters

You can configure the BFD session global parameters for all BFD sessions on the device. Different BFD session parameters for each session can be achieved using the per session configuration commands.

**Before you begin**

Enable the BFD feature.

**SUMMARY STEPS**

1. configure terminal
2. [no] bfd multihop interval milliseconds min_rx milliseconds multiplier interval-multiplier
3. end

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
</tbody>
</table>
| **Example:**  
  switch# configure terminal  
  switch(config)# | |
| **Step 2** [no] bfd multihop interval milliseconds min_rx milliseconds multiplier interval-multiplier | Configures the BFD multihop session global parameters for all BFD sessions on the device. This command overrides |
Configuring Bidirectional Forwarding Detection

Configuring Per Multihop Session BFD Parameters

You can configure per multihop session BFD parameters.

Before you begin
Enable the BFD feature. See the Enabling the BFD Feature section.

SUMMARY STEPS

1. configure terminal
2. router bgp as-number
3. neighbor (ip-address | ipv6-address) remote-as as-number
4. update-source interface
5. bfd
6. bfd multihop interval mintx min_rx msec multiplier value
7. bfd multihop authentication keyed-sha1 keyid key ascii_key
8. copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters configuration mode.</td>
</tr>
</tbody>
</table>
| Example:
switch# configure terminal
switch(config)# | |
| **Step 2** router bgp as-number | Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. |
| Example:
switch(config)# router bgp 64496
switch(config-router)# | |
| **Step 3** neighbor (ip-address | ipv6-address) remote-as as-number | Configures the IPv4 or IPv6 address and AS number for a remote BGP peer. The ip-address format is x.x.x.x. The ipv6-address format is A:B::C:D. |
| Example:
switch(config-router)# neighbor 209.165.201.1 remote-as 64497
switch(config-router-neighbor)# | |
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td>update-source interface</td>
<td>Retrieves the source IP address of the BFD session from the interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-router-neighbor)# update-source Ethernet1/4</td>
<td></td>
</tr>
<tr>
<td>switch(config-router-neighbor)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
<tr>
<td>bfd</td>
<td>Enables BFD for this BGP peer.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-router-neighbor)# bfd</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td></td>
</tr>
<tr>
<td>bfd multihop interval mintx min_rx msec multiplier value</td>
<td>Configures Multihop BFD interval values for this neighbor. The $mintx$ and $msec$ range is from 250 to 999 milliseconds and the default is 250. The multiplier range is from 1 to 50. The multiplier default is 3.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-router-neighbor)# bfd multihop interval 250 min_rx 250 multiplier 3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td></td>
</tr>
<tr>
<td>bfd multihop authentication keyed-sha1 keyid id key ascii_key</td>
<td>(Optional) Configures SHA-1 authentication for BFDs on Multihop BFD session over this neighbor. The $ascii_key$ string is a secret key shared among BFD peers. The $id$ value, a number between 0 and 255, is assigned to this particular $ascii_key$. BFD packets specify the key by $id$, allowing the use of multiple active keys. To disable SHA-1 authentication on the interface, use the no form of the command.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-router-neighbor)# bfd multihop authentication keyed-sha1 keyid 1 ascii_key cisco123</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td></td>
</tr>
<tr>
<td>copy running-config startup-config</td>
<td>(Optional) Saves the configuration change.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-router-neighbor)# copy running-config startup-config</td>
<td></td>
</tr>
</tbody>
</table>

### Configuration Examples for BFD

This example shows how to configure BFD for OSPFv2 on Ethernet 2/1, using the default BFD session parameters:

```
feature bfd
feature ospf
router ospf Test1
interface ethernet 2/1
ip ospf bfd
no shutdown
```

This example shows how to configure BFD for all EIGRP interfaces, using the default BFD session parameters:

```
feature bfd
feature eigrp
bfd interval 100 min_rx 100 multiplier 4
```
This example shows how to configure BFDv6:

```plaintext
feature bfd
feature ospfv3
router ospfv3 Test1
interface Ethernet2/7
  ipv6 router ospfv3 Test1 area 0.0.0.0
  ospfv3 bfd
  no shutdown
```

### Show Example for BFD

This example shows results of the `show bfd ipv6 neighbors details` command.

```
#show bfd ipv6 neighbors details

OurAddr               NeighAddr            RD/DR  Mult State  Int
cc:10::2              cc:10::1             1090519335/1090519260 Up 5692(3) Up Po1

Default

Session state is Up and using echo function with 250 ms interval
Local Diag: 0, Demand mode: 0, Poll bit: 0, Authentication: None
MinTxInt: 250000 us, MinRxInt: 2000000 us, Multiplier: 3
Received MinRxInt: 2000000 us, Received Multiplier: 3
Holdown (hits): 6000 ms (4), Hello (hits): 2000 ms (205229)
Rx Count: 227965, Rx Interval (ms) min/max/avg: 124/1520/1510 last: 307 ms ago
Tx Count: 205229, Tx Interval (ms) min/max/avg: 1677/1677/1677 last: 587 ms ago
Registered protocols: BGP
Uptime: 3 days 23 hrs 31 mins 13 secs
Last packet: Version: 1 - Diagnostic: 0
  State bit: Up - Demand bit: 0
  Poll bit: 0 - Final bit: 0
  Multiplier: 3 - Length: 24
  My Discr.: 1090519260 - Your Discr.: 1090519335
  Min tx interval: 250000 - Min rx interval: 2000000
  Min Echo interval: 250000 - Authentication bit: 0
Hosting LC: 1, Down reason: None, Reason not-hosted: None
```

### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>BFD commands</td>
<td><em>Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide</em></td>
</tr>
</tbody>
</table>
### RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 5880</td>
<td>Bidirectional Forwarding Detection (BFD)</td>
</tr>
<tr>
<td>RFC 5881</td>
<td>BFD for IPv4 and IPv6 (Single Hop)</td>
</tr>
<tr>
<td>RFC 7130</td>
<td>Bidirectional Forwarding Detection (BFD) on Link Aggregation Group (LAG) Interfaces</td>
</tr>
</tbody>
</table>
CHAPTER 7

Configuring Port Channels

- About Port Channels, on page 167
- Port Channels, on page 168
- Port-Channel Interfaces, on page 168
- Basic Settings, on page 169
- Compatibility Requirements, on page 170
- Load Balancing Using Port Channels, on page 172
- Symmetric Hashing, on page 173
- Guidelines and Limitations for ECMP, on page 173
- Resilient Hashing, on page 174
- GTP Tunnel Load Balancing, on page 174
- LACP, on page 175
- Licensing Requirements for Port Channeling, on page 181
- Prerequisites for Port Channeling, on page 181
- Guidelines and Limitations, on page 181
- Default Settings, on page 183
- Configuring Port Channels, on page 183

About Port Channels

A port channel is an aggregation of multiple physical interfaces that creates a logical interface. You can bundle up to 32 individual active links into a port channel to provide increased bandwidth and redundancy. Port channeling also load balances traffic across these physical interfaces. The port channel stays operational as long as at least one physical interface within the port channel is operational.

You can create a Layer 2 port channel by bundling compatible Layer 2 interfaces, or you can create Layer 3 port channels by bundling compatible Layer 3 interfaces. You cannot combine Layer 2 and Layer 3 interfaces in the same port channel.

You can also change the port channel from Layer 3 to Layer 2. See the Configuring Layer 2 Interfaces chapter for information about creating Layer 2 interfaces.

Any configuration changes that you apply to the port channel are applied to each member interface of that port channel. For example, if you configure Spanning Tree Protocol (STP) parameters on the port channel, the Cisco NX-OS software applies those parameters to each interface in the port channel.
After a Layer 2 port becomes part of a port channel, all switchport configurations must be done on the port channel; you can no longer apply switchport configurations to individual port-channel members. You cannot apply Layer 3 configurations to an individual port-channel member either; you must apply the configuration to the entire port channel.

You can use static port channels, with no associated aggregation protocol, for a simplified configuration. For more flexibility, you can use the Link Aggregation Control Protocol (LACP), which is defined in IEEE 802.3ad. When you use LACP, the link passes protocol packets. You cannot configure LACP on shared interfaces.

See the LACP Overview section for information about LACP.

Port Channels

A port channel bundles physical links into a channel group to create a single logical link that provides the aggregate bandwidth of up to 32 physical links. If a member port within a port channel fails, the traffic previously carried over the failed link switches to the remaining member ports within the port channel.

However, you can enable the LACP to use port channels more flexibly. Configuring port channels with LACP and static port channels require a slightly different procedure (see the “Configuring Port Channels” section).

The device does not support Port Aggregation Protocol (PAgP) for port channels.

Each port can be in only one port channel. All the ports in a port channel must be compatible; they must use the same speed and duplex mode (see the “Compatibility Requirements” section). When you run static port channels with no aggregation protocol, the physical links are all in the on channel mode; you cannot change this mode without enabling LACP (see the “Port-Channel Modes” section).

You can create port channels directly by creating the port-channel interface, or you can create a channel group that acts to aggregate individual ports into a bundle. When you associate an interface with a channel group, the software creates a matching port channel automatically if the port channel does not already exist. In this instance, the port channel assumes the Layer 2 or Layer 3 configuration of the first interface. You can also create the port channel first. In this instance, the Cisco NX-OS software creates an empty channel group with the same channel number as the port channel and takes the default Layer 2 or Layer 3 configuration, as well as the compatibility configuration (see the “Compatibility Requirements” section).

The port channel is operationally up when at least one of the member ports is up and that port’s status is channeling. The port channel is operationally down when all member ports are operationally down.

Port-Channel Interfaces

The following shows port-channel interfaces.
You can classify port-channel interfaces as Layer 2 or Layer 3 interfaces. In addition, you can configure Layer 2 port channels in either access or trunk mode. Layer 3 port-channel interfaces have routed ports as channel members.

You can configure a Layer 3 port channel with a static MAC address. If you do not configure this value, the Layer 3 port channel uses the router MAC of the first channel member to come up. See the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for information about configuring static MAC addresses on Layer 3 port channels.

See the "Configuring Layer 2 Interfaces" chapter for information about configuring Layer 2 ports in access or trunk mode and the "Configuring Layer 3 Interfaces" chapter for information about configuring Layer 3 interfaces and subinterfaces.

**Basic Settings**

You can configure the following basic settings for the port-channel interface:

- **Bandwidth**—Use this setting for informational purposes only; this setting is to be used by higher-level protocols.
- **Delay**—Use this setting for informational purposes only; this setting is to be used by higher-level protocols.
- **Description**
- **Duplex**
- **IP addresses**
- **Maximum Transmission Unit (MTU)**
- **Shutdown**
Compatibility Requirements

When you add an interface to a channel group, the software checks certain interface attributes to ensure that the interface is compatible with the channel group. For example, you cannot add a Layer 3 interface to a Layer 2 channel group. The Cisco NX-OS software also checks a number of operational attributes for an interface before allowing that interface to participate in the port-channel aggregation.

The compatibility check includes the following operational attributes:

- Network layer
- (Link) speed capability
- Speed configuration
- Duplex capability
- Duplex configuration
- Port mode
- Access VLAN
- Trunk native VLAN
- Tagged or untagged
- Allowed VLAN list
- MTU size
- SPAN—Cannot be a SPAN source or a destination port
- Storm control
- Flow-control capability
- Flow-control configuration
- Media type, either copper or fiber

Use the `show port-channel compatibility-parameters` command to see the full list of compatibility checks that the Cisco NX-OS uses.

You can only add interfaces configured with the channel mode set to on to static port channels, and you can only add interfaces configured with the channel mode as active or passive to port channels that are running LACP. You can configure these attributes on an individual member port. If you configure a member port with an incompatible attribute, the software suspends that port in the port channel.

Alternatively, you can force ports with incompatible parameters to join the port channel if the following parameters are the same:

- (Link) speed capability
- Speed configuration
• Duplex capability
• Duplex configuration
• Flow-control capability
• Flow-control configuration

When the interface joins a port channel, some of its individual parameters are removed and replaced with the values on the port channel as follows:

• Bandwidth
• Delay
• Extended Authentication Protocol over UDP
• VRF
• IP address
• MAC address
• Spanning Tree Protocol
• NAC
• Service policy
• Access control lists (ACLs)

Many interface parameters remain unaffected when the interface joins or leaves a port channel as follows:

• Beacon
• Description
• CDP
• LACP port priority
• Debounce
• UDLD
• MDIX
• Rate mode
• Shutdown
• SNMP trap

---

**Note**

When you delete the port channel, the software sets all member interfaces as if they were removed from the port channel.

See the “LACP Marker Responders” section for information about port-channel modes.
Load Balancing Using Port Channels

The Cisco NX-OS software load balances traffic across all operational interfaces in a port channel by hashing the addresses in the frame to a numerical value that selects one of the links in the channel. Port channels provide load balancing by default. Port-channel load balancing uses MAC addresses, IP addresses, or Layer 4 port numbers to select the link. Port-channel load balancing uses either source or destination addresses or ports, or both source and destination addresses or ports.

You can configure the load-balancing mode to apply to all port channels that are configured on the entire device. You can configure one load-balancing mode for the entire device. You cannot configure the load-balancing method per port channel.

You can configure the type of load-balancing algorithm used. You can choose the load-balancing algorithm that determines which member port to select for egress traffic by looking at the fields in the frame.

The default load-balancing mode for Layer 3 interfaces is the source and destination IP L4 ports, and the default load-balancing mode for non-IP traffic is the source and destination MAC address. Use the `port-channel load-balance` command to set the load-balancing method among the interfaces in the channel-group bundle. The default method for Layer 2 packets is src-dst-mac. The default method for Layer 3 packets is src-dst ip-l4port.

You can configure the device to use one of the following methods to load balance across the port channel:

- Destination MAC address
- Source MAC address
- Source and destination MAC address
- Destination IP address
- Source IP address
- Source and destination IP address
- Source TCP/UDP port number
- Destination TCP/UDP port number
- Source and destination TCP/UDP port number
- GRE inner IP headers with source, destination and source-destination

Non-IP and Layer 3 port channels both follow the configured load-balancing method, using the source, destination, or source and destination parameters. For example, when you configure load balancing to use the source IP address, all non-IP traffic uses the source MAC address to load balance the traffic while the Layer 3 traffic load balances the traffic using the source IP address. Similarly, when you configure the destination MAC address as the load-balancing method, all Layer 3 traffic uses the destination IP address while the non-IP traffic load balances using the destination MAC address.

**Note** Configuring hash load balancing applies to unicast and multicast traffic on Cisco Nexus 9200 and 9300-EX Series switches.
The unicast and multicast traffic is load-balanced across port-channel links based on configured load-balancing algorithm displayed in `show port-channel load-balancing` command output.

The multicast traffic uses the following methods for load balancing with port channels:

- Multicast traffic with Layer 4 information—Source IP address, source port, destination IP address, destination port
- Multicast traffic without Layer 4 information—Source IP address, destination IP address
- Non-IP multicast traffic—Source MAC address, destination MAC address

**Note**

Devices that run Cisco IOS can optimize the behavior of the member ports ASICs if a failure of a single member occurred by running the port-channel hash-distribution command. The Cisco Nexus 9000 Series device performs this optimization by default and does not require or support this command. Cisco NX-OS does support the customization of the load-balancing criteria on port channels through the port-channel load-balance command for the entire device.

**Symmetric Hashing**

To be able to effectively monitor traffic on a port channel, it is essential that each interface connected to a port channel receives both forward and reverse traffic flows. Normally, there is no guarantee that the forward and reverse traffic flows will use the same physical interface. However, when you enable symmetric hashing on the port channel, bidirectional traffic is forced to use the same physical interface and each physical interface in the port channel is effectively mapped to a set of flows.

When symmetric hashing is enabled, the parameters used for hashing, such as the source and destination IP address, are normalized before they are entered into the hashing algorithm. This process ensures that when the parameters are reversed (the source on the forward traffic becomes the destination on the reverse traffic), the hash output is the same. Therefore, the same interface is chosen.

Only the following load-balancing algorithms support symmetric hashing:

- `src-dst ip`
- `src-dst ip-l4port`

**Guidelines and Limitations for ECMP**

You might observe that load balancing with Layer 2/Layer 3 GW flows are not load balanced equally among all links when the switch comes up initially after reload. There are two CLIs to change the ECMP hash configuration in the hardware. The two CLI commands are mutually exclusive.

- Enter the `port-channel load-balance [src | src-dst | dst] mac` command for MAC-based only hash.
- For hash based on IP/Layer 4 ports, enter either the `ip load-share` or `port-channel load-balance` command.
- The `port-channel load-balance` command can overwrite the `ip load-share` command. It is better to enter the `port-channel load-balance` command which helps to set both the IP and MAC parameters.
Resilient Hashing

With the exponential increase in the number of physical links used in data centers, there is also the potential for an increase in the number of failed physical links. In static hashing systems that are used for load balancing flows across members of port channels or Equal Cost Multipath (ECMP) groups, each flow is hashed to a link. If a link fails, all flows are rehashed across the remaining working links. This rehashing of flows to links results in some packets being delivered out of order even for those flows that were not hashed to the failed link.

This rehashing also occurs when a link is added to the port channel or Equal Cost Multipath (ECMP) group. All flows are rehashed across the new number of links, which results in some packets being delivered out of order. Resilient hashing supports only unicast traffic.

Resilient hashing maps flows to physical ports. In case a link fails, the flows assigned to the failed link are redistributed uniformly among the working links. The existing flows through the working links are not rehashed and their packets are not delivered out of order.

Resilient hashing is supported only for ECMP groups and not on port channel interfaces. Resiliency is guaranteed only upon deletion of a member in an ECMP group, not on adding a member to the ECMP group. When a link is added to the port channel or ECMP group, some of the flows hashed to the existing links are rehashed to the new link, but not across all existing links.

Resilient hashing also occurs when a link is added to the port channel or Equal Cost Multipath (ECMP) group. Resilient hashing is supported on both Layer 2 port-channel member links and Layer 3 ECMP paths on the routing table.

Resilient hashing supports IPv4 and IPv6 known unicast traffic, but it does not support IPv4 multicast traffic.

Note
Resilient hashing is supported on Network Forwarding Engine (NFE) based Cisco Nexus 9300 platform switches and Cisco Nexus 9500 platform switches.

GTP Tunnel Load Balancing

GPRS Tunneling Protocol (GTP) is used mainly to deliver mobile data on wireless networks via Cisco Nexus 9000 Series switches as the core router. When two routers carrying GTP traffic are connected with link bundling, the traffic is required to be distributed evenly between all bundle members.

To achieve load balancing, Cisco Nexus 9000 Series switches use 5-tuple load balancing mechanism. The load balancing mechanism takes into account the source IP, destination IP, protocol, Layer 4 resource and destination port (if traffic is TCP or UDP) fields from the packet. In the case of GTP traffic, a limited number of unique values for these fields restrict the equal distribution of traffic load on the tunnel.

In order to avoid polarization for GTP traffic in load balancing, a tunnel endpoint identifier (TEID) in the GTP header is used instead of a UDP port number. Since the TEID is unique per tunnel, traffic can be evenly load balanced across multiple links in the bundle.
Beginning Cisco Nexus Release 7.0(3)I7(4), GTP Tunnel Load Balancing is supported on Cisco Nexus 9300-FX and 9364C platform switches.

GTP Tunnel Load Balancing is supported on Cisco Nexus 9300-EX, 9300-FX, 9300-FX2 and 9364C platform switches.

This feature overrides the source and destination port information with the 32-bit TEID value that is present in GTPU packets.

GTP tunnel load balancing feature adds support for:
- GTP with IPv4/IPv6 transport header on physical interface
- GTP traffic over TE tunnel
- GTPU with UDP port 2152

The `ip load-sharing address source-destination gtpu` command enables the GTP tunnel load balancing.

To know the egress interface for GTP traffic after load balancing, use `show cef {ipv4 | ipv6} exact-route` command with TEID in place of L4 protocol source and destination port number. Use 16MSBis of TEID in source port and 16LSBits of TEID in destination port.

### LACP

LACP allows you to configure up to 16 interfaces into a port channel.

#### LACP Overview

The Link Aggregation Control Protocol (LACP) for Ethernet is defined in IEEE 802.1AX and IEEE 802.3ad. This protocol controls how physical ports are bundled together to form one logical channel.

---

**Note**

You must enable LACP before you can use LACP. By default, LACP is disabled. See the “Enabling LACP” section for information about enabling LACP.

The system automatically takes a checkpoint before disabling the feature, and you can roll back to this checkpoint. See the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide for information about rollbacks and checkpoints.

The following figure shows how individual links can be combined into LACP port channels and channel groups as well as function as individual links.
With LACP, you can bundle up to 32 interfaces in a channel group.

**Note**
When you delete the port channel, the software automatically deletes the associated channel group. All member interfaces revert to their original configuration.

**Note**
If you downgrade a Cisco Nexus 9500 series switch that is configured to use LACP vPC convergence feature, that runs Cisco NX-OS Release 7.0(3)I7(5) to a lower release, the configuration is removed. You must configure the LACP vPC convergence feature again when you upgrade the switch.

You cannot disable LACP while any LACP configurations are present.

### Port-Channel Modes

Individual interfaces in port channels are configured with channel modes. When you run static port channels with no aggregation protocol, the channel mode is always set to `on`.

After you enable LACP globally on the device, you enable LACP for each channel by setting the channel mode for each interface to `active` or `passive`. You can configure either channel mode for individual links in the LACP channel group when you are adding the links to the channel group.

**Note**
You must enable LACP globally before you can configure an interface in either the `active` or `passive` channel mode.

The following table describes the channel modes.
Table 10: Channel Modes for Individual Links in a Port Channel

<table>
<thead>
<tr>
<th>Channel Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>passive</td>
<td>LACP mode that places a port into a passive negotiating state in which the port responds to LACP packets that it receives but does not initiate LACP negotiation.</td>
</tr>
<tr>
<td>active</td>
<td>LACP mode that places a port into an active negotiating state in which the port initiates negotiations with other ports by sending LACP packets.</td>
</tr>
<tr>
<td>on</td>
<td>All static port channels (that are not running LACP) remain in this mode. If you attempt to change the channel mode to active or passive before enabling LACP, the device displays an error message. You enable LACP on each channel by configuring the interface in that channel for the channel mode as either active or passive. When an LACP attempts to negotiate with an interface in the on state, it does not receive any LACP packets and becomes an individual link with that interface; it does not join the LACP channel group. The default port-channel mode is on.</td>
</tr>
</tbody>
</table>

Both the passive and active modes allow LACP to negotiate between ports to determine if they can form a port channel based on criteria such as the port speed and the trunking state. The passive mode is useful when you do not know whether the remote system, or partner, supports LACP.

Ports can form an LACP port channel when they are in different LACP modes if the modes are compatible as in the following examples:

- A port in active mode can form a port channel successfully with another port that is in active mode.
- A port in active mode can form a port channel with another port in passive mode.
- A port in passive mode cannot form a port channel with another port that is also in passive mode, because neither port will initiate negotiation.
- A port in on mode is not running LACP and cannot form a port channel with another port that is in active or passive mode.

**LACP ID Parameters**

This section describes the LACP parameters.

**LACP System Priority**

Each system that runs LACP has an LACP system priority value. You can accept the default value of 32768 for this parameter, or you can configure a value between 1 and 65535. LACP uses the system priority with
the MAC address to form the system ID and also uses the system priority during negotiation with other devices. A higher system priority value means a lower priority.

---

**Note**
The LACP system ID is the combination of the LACP system priority value and the MAC address.

---

**LACP Port Priority**

Each port that is configured to use LACP has an LACP port priority. You can accept the default value of 32768 for the LACP port priority, or you can configure a value between 1 and 65535. LACP uses the port priority with the port number to form the port identifier.

LACP uses the port priority to decide which ports should be put in standby mode when there is a limitation that prevents all compatible ports from aggregating and which ports should be put into active mode. A higher port priority value means a lower priority for LACP. You can configure the port priority so that specified ports have a lower priority for LACP and are most likely to be chosen as active links, rather than hot-standby links.

**LACP Administrative Key**

LACP automatically configures an administrative key value equal to the channel-group number on each port configured to use LACP. The administrative key defines the ability of a port to aggregate with other ports. A port’s ability to aggregate with other ports is determined by these factors:

- Port physical characteristics, such as the data rate and the duplex capability
- Configuration restrictions that you establish

**LACP Marker Responders**

You can dynamically redistribute the data traffic by using port channels. This redistribution might result from a removed or added link or a change in the load-balancing scheme. Traffic redistribution that occurs in the middle of a traffic flow can cause misordered frames.

LACP uses the Marker Protocol to ensure that frames are not duplicated or reordered due to this redistribution. The Marker Protocol detects when all the frames of a given traffic flow are successfully received at the remote end. LACP sends Marker PDUs on each of the port-channel links. The remote system responds to the Marker PDU once it receives all the frames received on this link prior to the Marker PDU. The remote system then sends a Marker Responder. Once the Marker Responders are received by the local system on all member links of the port channel, the local system can redistribute the frames in the traffic flow with no chance of misordering. The software supports only Marker Responders.

**LACP-Enabled and Static Port Channels Differences**

The following table summarizes the major differences between port channels with LACP enabled and static port channels.
Table 11: Port Channels with LACP Enabled and Static Port Channels

<table>
<thead>
<tr>
<th>Configurations</th>
<th>Port Channels with LACP Enabled</th>
<th>Static Port Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol applied</td>
<td>Enable globally</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Channel mode of links</td>
<td>Can be either:</td>
<td>Can only be On</td>
</tr>
<tr>
<td></td>
<td>• Active</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Passive</td>
<td></td>
</tr>
<tr>
<td>Maximum number of links</td>
<td>32</td>
<td>32</td>
</tr>
</tbody>
</table>

**LACP Compatibility Enhancements**

When a Cisco Nexus 9000 Series device is connected to a non-Nexus peer, its graceful failover defaults may delay the time taken for a disabled port to be brought down or cause traffic from the peer to be lost. To address these conditions, the `lacp graceful-convergence` command was added.

By default, LACP sets a port to the suspended state if it does not receive an LACP PDU from the peer. In some cases, although this feature helps in preventing loops created due to misconfigurations, it can cause servers to fail to boot up because they require LACP to logically bring up the port. You can put a port into an individual state by using the `lacp suspend-individual`.

**Delayed LACP**

LACP port-channels exchange LACP PDUs for quick bundling of links when connecting a server and a switch. However, the links go into suspended state when the PDUs are not received.

The delayed LACP feature enables one port-channel member, the delayed-LACP port, to come up first as a member of a regular port-channel before LACP PDUs are received. After it is connected in LACP mode, other members, the auxiliary LACP ports, are brought up. This avoids having the links becoming suspended when PDUs are not received.

**LACP Port-Channel Minimum Links and MaxBundle**

A port channel aggregates similar ports to provide increased bandwidth in a single manageable interface.

The introduction of the minimum links and maxbundle feature further refines LACP port-channel operation and provides increased bandwidth in one manageable interface.

The LACP port-channel minimum links feature does the following:

- Configures the minimum number of ports that must be linked up and bundled in the LACP port channel.
- Prevents the low-bandwidth LACP port channel from becoming active.
- Causes the LACP port channel to become inactive if there are few active members ports to supply the required minimum bandwidth.

The LACP MaxBundle defines the maximum number of bundled ports allowed in a LACP port channel.
The LACP MaxBundle feature does the following:

- Defines an upper limit on the number of bundled ports in an LACP port channel.
- Allows hot-standby ports with fewer bundled ports. (For example, in an LACP port channel with five ports, you can designate two of those ports as hot-standby ports.)

**Note**
The minimum links and maxbundle feature works only with LACP port channels. However, the device allows you to configure this feature in non-LACP port channels, but the feature is not operational.

### LACP Fast Timers

You can change the LACP timer rate to modify the duration of the LACP timeout. Use the lacp rate command to set the rate at which LACP control packets are sent to an LACP-supported interface. You can change the timeout rate from the default rate (30 seconds) to the fast rate (1 second). This command is supported only on LACP-enabled interfaces. To configure the LACP fast time rate, see the “Configuring the LACP Fast Timer Rate” section.

ISSU and ungraceful switchovers are not supported with LACP fast timers.

### Virtualization Support

You must configure the member ports and other port channel-related configuration from the virtual device context (VDC) that contains the port channel and member ports. You can use the numbers from 1 to 4096 in each VDC to number the port channels.

All ports in one port channel must be in the same VDC. When you are using LACP, all possible 8 active ports and all possible 8 standby ports must be in the same VDC.

**Note**
You must configure load balancing using port channels in the default VDC. See the “Load Balancing Using Port Channels” section for more information about load balancing.

### High Availability

Port channels provide high availability by load balancing traffic across multiple ports. If a physical port fails, the port channel is still operational if there is an active member in the port channel. You can bundle ports from different modules and create a port channel that remains operational even if a module fails because the settings are common across the module.

Port channels support stateful and stateless restarts. A stateful restart occurs on a supervisor switchover. After the switchover, the Cisco NX-OS software applies the runtime configuration after the switchover.

The port channel goes down if the operational ports fall below the configured minimum links number.
See the *Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide* for complete information about high-availability features.

### Licensing Requirements for Port Channeling

The following table shows the licensing requirements for this feature:

<table>
<thead>
<tr>
<th>Product</th>
<th>License Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco NX-OS</td>
<td>Port channeling requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS image and is provided at no extra charge to you.</td>
</tr>
</tbody>
</table>

### Prerequisites for Port Channeling

Port channeling has the following prerequisites:

- You must be logged onto the device.
- All ports for a single port channel must be either Layer 2 or Layer 3 ports.
- All ports for a single port channel must meet the compatibility requirements. See the “Compatibility Requirements” section for more information about the compatibility requirements.
- You must configure load balancing from the default VDC.

### Guidelines and Limitations

Port channeling has the following configuration guidelines and limitations:

- `show` commands with the `internal` keyword are not supported.
- The LACP port-channel minimum links and maxbundle feature is not supported for host interface port channels.
- Enable LACP before you can use that feature.
- You can configure multiple port channels on a device.
- Do not put shared and dedicated ports into the same port channel. (See the “Configuring Basic Interface Parameters” chapter for information about shared and dedicated ports.)
- For Layer 2 port channels, ports with different STP port path costs can form a port channel if they are compatibly configured with each other. See the “Compatibility Requirements” section for more information about the compatibility requirements.
- In STP, the port-channel cost is based on the aggregated bandwidth of the port members.
After you configure a port channel, the configuration that you apply to the port channel interface affects the port channel member ports. The configuration that you apply to the member ports affects only the member port where you apply the configuration.

LACP does not support half-duplex mode. Half-duplex ports in LACP port channels are put in the suspended state.

Do not configure ports that belong to a port channel group as private VLAN ports. While a port is part of the private VLAN configuration, the port channel configuration becomes inactive.

Channel member ports cannot be a source or destination SPAN port.

Port-channels are not supported on generation 1 100G line cards (N9K-X9408PC-CFP2) or generic expansion modules (N9K-M4PC-CFP2).

Port-channels are supported on devices with generation 2 (and later) 100G interfaces.

The port channel might be affected by the limitations of the Application Leaf Engine (ALE) uplink ports on Cisco Nexus 9300 and 9500 Series devices: Limitations for ALE Uplink Ports.

Resilient hashing for port channels is not supported on Cisco Nexus 9200, Cisco Nexus 9300-EX, and Cisco Nexus 9500 switches with 9700-EX line cards.

Resilient hashing (port-channel load-balancing resiliency) and VXLAN configurations are not compatible with VTEPs using ALE uplink ports.

Resilient hashing is disabled by default.

The maximum number of subinterfaces for a port is 511. The maximum number of subinterfaces for a satellite/FEX port is 63.

For a given set of ports in the same quadrant, you must use a transceiver of the same speed. You should not mix speeds within a set of ports in the same quadrant. The port numbers that share same quadrant are as follows:

- 1, 4, 7, 10
- 2, 5, 8, 11
- 3, 6, 9, 12
- 13, 16, 19, 22
- 14, 17, 20, 23
- 15, 18, 21, 24
- 25, 28, 31, 34
- 26, 29, 32, 35
- 27, 30, 33, 36
- 37, 40, 43, 46
- 38, 41, 44, 47
• 39,42,45,48


Default Settings

The following table lists the default settings for port-channel parameters.

Table 12: Default Port-Channel Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port channel</td>
<td>Admin up</td>
</tr>
<tr>
<td>Load balancing method for Layer 3 interfaces</td>
<td>Source and destination IP address</td>
</tr>
<tr>
<td>Load balancing method for Layer 2 interfaces</td>
<td>Source and destination MAC address</td>
</tr>
<tr>
<td>Load balancing per module</td>
<td>Disabled</td>
</tr>
<tr>
<td>LACP</td>
<td>Disabled</td>
</tr>
<tr>
<td>Channel mode</td>
<td>on</td>
</tr>
<tr>
<td>LACP system priority</td>
<td>32768</td>
</tr>
<tr>
<td>LACP port priority</td>
<td>32768</td>
</tr>
<tr>
<td>Minimum links for LACP</td>
<td>1</td>
</tr>
<tr>
<td>Maxbundle</td>
<td>32</td>
</tr>
<tr>
<td>Minimum links for FEX fabric port channel</td>
<td>1</td>
</tr>
</tbody>
</table>

Configuring Port Channels

See the "Configuring Basic Interface Parameters” chapter for information about configuring the maximum transmission unit (MTU) for the port-channel interface. See the “Configuring Layer 3 Interfaces” chapter for information about configuring IPv4 and IPv6 addresses on the port-channel interface.

Note

If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.
Creating a Port Channel

You can create a port channel before you create a channel group. The software automatically creates the associated channel group.

**Note**

When the port channel is created before the channel group, the port channel should be configured with all of the interface attributes that the member interfaces are configured with. Use the `switchport mode trunk {allowed vlan vlan-id | native vlan-id}` command to configure the members.

This is required only when the channel group members are Layer 2 ports (switchport) and trunks (switchport mode trunk).

**Note**

Use the `no interface port-channel` command to remove the port channel and delete the associated channel group.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>no interface port-channel channel-number</td>
<td>Removes the port channel and deletes the associated channel group.</td>
</tr>
</tbody>
</table>

**Example:**

```bash
switch(config)# no interface port-channel 1
```

**Before you begin**

Enable LACP if you want LACP-based port channels.

**SUMMARY STEPS**

1. configure terminal
2. interface port-channel channel-number
3. show port-channel summary
4. no shutdown
5. copy running-config startup-config

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**
  configure terminal
  Example:
  ```bash
  switch# configure terminal
  switch(config)#
  ```
| Enters global configuration mode. |
| **Step 2**
  interface port-channel channel-number
  Example: | Specifies the port-channel interface to configure, and enters the interface configuration mode. The range is from 1 to 4096. The Cisco NX-OS software automatically creates the channel group if it does not already exist. |
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| `switch(config)# interface port-channel 1`  
`switch(config-if)` | |

<table>
<thead>
<tr>
<th>Step 3</th>
<th>show port-channel summary</th>
<th>(Optional) Displays information about the port channel.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>switch(config-router)# show port-channel summary</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th><code>no shutdown</code></th>
<th>(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.</th>
</tr>
</thead>
</table>
| Example: | `switch# configure terminal`  
`switch(config)# int e3/1`  
`switch(config-if)# no shutdown` | |

<table>
<thead>
<tr>
<th>Step 5</th>
<th><code>copy running-config startup-config</code></th>
<th>(Optional) Copies the running configuration to the startup configuration.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>switch(config)# copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>

### Example

This example shows how to create a port channel:

```
switch# configure terminal
switch (config)# interface port-channel 1
```

See the “Compatibility Requirements” section for details on how the interface configuration changes when you delete the port channel.

### Adding a Layer 2 Port to a Port Channel

You can add a Layer 2 port to a new channel group or to a channel group that already contains Layer 2 ports. The software creates the port channel associated with this channel group if the port channel does not already exist.

#### Note

Use the `no channel-group` command to remove the port from the channel group.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>no channel-group</code></td>
<td>Removes the port from the channel group.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>switch(config)# no channel-group</code></td>
<td></td>
</tr>
</tbody>
</table>

### Before you begin

Enable LACP if you want LACP-based port channels.
All Layer 2 member ports must run in full-duplex mode and at the same speed

**SUMMARY STEPS**

1. configure terminal
2. interface type slot/port
3. switchport
4. switchport mode trunk
5. switchport trunk {allowed vlan vlan-id | native vlan-id}
6. channel-group channel-number [force] [mode {on | active | passive}]
7. show interface type slot/port
8. no shutdown
9. copy running-config startup-config

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
</tbody>
</table>
| **Example:**      | switch# configure terminal  
switch(config)# | |
| **Step 2**        | interface type slot/port | Specifies the interface that you want to add to a channel group, and enters the interface configuration mode. |
| **Example:**      | switch(config)# interface ethernet 1/4  
switch(config-if)# | |
| **Step 3**        | switchport | Configures the interface as a Layer 2 access port. |
| **Example:**      | switch(config)# switchport | |
| **Step 4**        | switchport mode trunk | (Optional) Configures the interface as a Layer 2 trunk port. |
| **Example:**      | switch(config)# switchport mode trunk | |
| **Step 5**        | switchport trunk {allowed vlan vlan-id | native vlan-id} | (Optional) Configures necessary parameters for a Layer 2 trunk port. |
| **Example:**      | switch(config)# switchport trunk native 3  
switch(config-if)# | |
| **Step 6**        | channel-group channel-number [force] [mode {on | active | passive}] | Configures the port in a channel group and sets the mode. The channel-number range is from 1 to 4096. This command creates the port channel associated with this channel group if the port channel does not already exist. All static port-channel interfaces are set to mode on. You must set all LACP-enabled port-channel interfaces to active or passive. The default mode is on. |
| **Example:**      | * switch(config-if)# channel-group 5  
* switch(config-if)# channel-group 5 force | |
### Command or Action

**Purpose**

(Optional) Forces an interface with some incompatible configurations to join the channel. The forced interface must have the same speed, duplex, and flow control settings as the channel group.

**Note**

The `force` option fails if the port has a QoS policy mismatch with the other members of the port channel.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td><code>show interface type slot/port</code></td>
<td>(Optional) Displays interface information.</td>
</tr>
<tr>
<td></td>
<td>Example: switch# show interface port channel 5</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td><code>no shutdown</code></td>
<td>(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.</td>
</tr>
<tr>
<td></td>
<td>Example: switch# configure terminal switch(config)# int e3/1 switch(config-if)# no shutdown</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td><code>copy running-config startup-config</code></td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td></td>
<td>Example: switch(config)# copy running-config startup-config</td>
<td></td>
</tr>
</tbody>
</table>

**Example**

This example shows how to add a Layer 2 Ethernet interface 1/4 to channel group 5:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# switchport
switch(config-if)# channel-group 5
```

### Adding a Layer 3 Port to a Port Channel

You can add a Layer 3 port to a new channel group or to a channel group that is already configured with Layer 3 ports. The software creates the port channel associated with this channel group if the port channel does not already exist.

If the Layer 3 port that you are adding has a configured IP address, the system removes that IP address before adding the port to the port channel. After you create a Layer 3 port channel, you can assign an IP address to the port-channel interface.
Use the `no channel-group` command to remove the port from the channel group. The port reverts to its original configuration. You must reconfigure the IP addresses for this port.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>no channel-group</code></td>
<td>Removes the port from the channel group.</td>
</tr>
</tbody>
</table>

**Example:**

```
switch(config)# no channel-group
```

**Before you begin**

Enable LACP if you want LACP-based port channels.

Remove any IP addresses configured on the Layer 3 interface.

**SUMMARY STEPS**

1. configure terminal
2. interface type slot/port
3. no switchport
4. channel-group channel-number [force] [mode {on | active | passive}]
5. show interface type slot/port
6. no shutdown
7. copy running-config startup-config

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>interface type slot/port</td>
<td>Specifies the interface that you want to add to a channel group, and enters the interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)# interface ethernet 1/4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>no switchport</td>
<td>Configures the interface as a Layer 3 port.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# no switchport</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>channel-group channel-number [force] [mode {on</td>
<td>active</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>* switch(config-if)# channel-group 5</td>
<td>(Optional) Forces an interface with some incompatible configurations to join the channel. The forced interface must have the same speed, duplex, and flow control settings as the channel group.</td>
</tr>
<tr>
<td>* switch(config-if)# channel-group 5 force</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>show interface type slot/port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>switch# show interface ethernet 1/4</td>
</tr>
<tr>
<td>(Optional) Displays interface information.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>no shutdown</th>
</tr>
</thead>
</table>
| Example: | switch# configure terminal  
switch(config)# int e3/1  
switch(config-if)# no shutdown |
| (Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state. | |

<table>
<thead>
<tr>
<th>Step 7</th>
<th>copy running-config startup-config</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>switch(config)# copy running-config startup-config</td>
</tr>
<tr>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
<td></td>
</tr>
</tbody>
</table>

### Example

This example shows how to add a Layer 3 Ethernet interface 1/5 to channel group 6 in on mode:

```plaintext
switch# configure terminal  
switch (config)# interface ethernet 1/5  
switch(config-if)# switchport  
switch(config-if)# channel-group 6
```

This example shows how to create a Layer 3 port-channel interface and assign the IP address:

```plaintext
switch# configure terminal  
switch (config)# interface port-channel 4  
switch(config-if)# ip address 192.0.2.1/8
```

### Configuring the Bandwidth and Delay for Informational Purposes

The bandwidth of the port channel is determined by the number of total active links in the channel.

You configure the bandwidth and delay on port-channel interfaces for informational purposes.

### SUMMARY STEPS

1. `configure terminal`
2. `interface port-channel channel-number`
3. `bandwidth value`
4. `delay value`
5. `exit`
6. `show interface port-channel channel-number`
7. `copy running-config startup-config`

### Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface port-channel channel-number</td>
<td>Specifies the port-channel interface that you want to configure, and enters the interface mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch(config)# interface port-channel 2</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config-if)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> bandwidth value</td>
<td>Specifies the bandwidth, which is used for informational purposes. The range is from 1 to 3,200,000,000 kbs. The default value depends on the total active interfaces in the channel group.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch(config-if)# bandwidth 60000000</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config-if)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> delay value</td>
<td>Specifies the throughput delay, which is used for informational purposes. The range is from 1 to 16,777,215 tens of microseconds. The default value is 10 microseconds.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch(config-if)# delay 10000</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config-if)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> exit</td>
<td>Exits the interface mode and returns to the configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch(config-if)# exit</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> show interface port-channel channel-number</td>
<td>(Optional) Displays interface information for the specified port channel.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch# show interface port-channel 2</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> copy running-config startup-config</td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch(config)# copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>

### Example

This example shows how to configure the informational parameters of the bandwidth and delay for port channel 5:

```
switch# configure terminal
switch (config)# interface port-channel 5
switch(config-if)# bandwidth 60000000
switch(config-if)# delay 10000
```
Shutting Down and Restarting the Port-Channel Interface

You can shut down and restart the port-channel interface. When you shut down a port-channel interface, no traffic passes and the interface is administratively down.

SUMMARY STEPS

1. configure terminal
2. interface port-channel channel-number
3. shutdown
4. exit
5. show interface port-channel channel-number
6. no shutdown
7. copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>switch# configure terminal switch(config)#</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>interface port-channel channel-number</td>
<td>Specifies the port-channel interface that you want to configure, and enters the interface mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config)# interface port-channel 2 switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>shutdown</td>
<td>Shuts down the interface. No traffic passes and the interface displays as administratively down. The default is no shutdown.</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config-if)# shutdown switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td></td>
<td>no shutdown</td>
<td>Use the no shutdown command to open the interface. The interface displays as administratively up. If there are no operational problems, traffic passes. The default is no shutdown.</td>
</tr>
<tr>
<td>Step 4</td>
<td>exit</td>
<td>Exits the interface mode and returns to the configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config-if)# exit switch(config)#</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>show interface port-channel channel-number</td>
<td>(Optional) Displays interface information for the specified port channel.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Configuring a Port-Channel Description

You can configure a description for a port channel.

**SUMMARY STEPS**

1. `configure terminal`
2. `interface port-channel channel-number`
3. `description`
4. `exit`
5. `show interface port-channel channel-number`
6. `copy running-config startup-config`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>configure terminal</code>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>switch# configure terminal&lt;br&gt;switch(config)#</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>interface port-channel channel-number</code>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>switch(config)# interface port-channel 2&lt;br&gt;switch(config-if)#</code></td>
</tr>
</tbody>
</table>
Configuring Port Channels

### Purpose

#### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 3</td>
<td>description</td>
<td>Allows you to add a description to the port-channel interface. You can use up to 80 characters in the description. By default, the description does not display; you must configure this parameter before the description displays in the output.</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config-if)# description engineering</td>
<td></td>
</tr>
</tbody>
</table>

#### Example:

| Step 4 | exit | Exits the interface mode and returns to the configuration mode. |
| Example: | switch(config-if)# exit |

#### Example:

| Step 5 | show interface port-channel channel-number | (Optional) Displays interface information for the specified port channel. |
| Example: | switch# show interface port-channel 2 |

#### Example:

| Step 6 | copy running-config startup-config | (Optional) Copies the running configuration to the startup configuration. |
| Example: | switch(config)# copy running-config startup-config |

### Example

This example shows how to add a description to port channel 2:

```
switch# configure terminal
switch (config)# interface port-channel 2
switch(config-if)# description engineering
```

### Configuring the Speed and Duplex Settings for a Port-Channel Interface

You can configure the speed and duplex settings for a port-channel interface.

#### SUMMARY STEPS

1. configure terminal
2. interface port-channel channel-number
3. speed {10 | 100 | 1000 | auto}
4. duplex {auto | full | half}
5. exit
6. show interface port-channel channel-number
7. copy running-config startup-config
## Configuring Load Balancing Using Port Channels

You can configure the load-balancing algorithm for port channels that applies to the entire device.

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface port-channel</td>
<td>Specifies the port-channel interface that you want to configure, and enters the interface mode.</td>
</tr>
<tr>
<td>channel-number</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# interface port-channel</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> speed</td>
<td>Sets the speed for the port-channel interface. The default is auto for autonegotiation.</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)# speed</td>
<td></td>
</tr>
<tr>
<td>auto</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> duplex</td>
<td>Sets the duplex for the port-channel interface. The default is auto for autonegotiation.</td>
</tr>
<tr>
<td>auto</td>
<td>full</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)# speed</td>
<td></td>
</tr>
<tr>
<td>auto</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> exit</td>
<td>Exits the interface mode and returns to the configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)# exit</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> show interface port-channel</td>
<td>(Optional) Displays interface information for the specified port channel.</td>
</tr>
<tr>
<td>channel-number</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# show interface port-channel</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> copy running-config startup-config</td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# copy running-config startup-config</td>
<td></td>
</tr>
</tbody>
</table>

### Example

This example shows how to set port channel 2 to 100 Mb/s:

```
switch# configure terminal
switch (config)# interface port-channel 2
switch(config-if)# speed 100
```
Use the **no port-channel load-balance** command to restore the default load-balancing algorithm of source-dest-mac for non-IP traffic and source-dest-ip for IP traffic.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>no port-channel load-balance</td>
<td>Restores the default load-balancing algorithm.</td>
</tr>
</tbody>
</table>

Example:
```
switch(config)# no port-channel load-balance
```

**Before you begin**

Enable LACP if you want LACP-based port channels.

**SUMMARY STEPS**

1. **configure terminal**
3. **show port-channel load-balance**
4. **copy running-config startup-config**

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)#</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>port-channel load-balance method</td>
<td>Specifies the load-balancing algorithm for the device. The range depends on the device. The default for Layer 3 is <strong>src-dst ip-l4port</strong> for both IPv4 and IPv6, and the default for non-IP is <strong>src-dst mac</strong>.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* switch(config)# port-channel load-balance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* switch(config)# src-dst mac</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* switch(config)#</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* switch(config)# no port-channel load-balance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* switch(config)# src-dst mac</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* switch(config)#</td>
<td></td>
</tr>
</tbody>
</table>
Enabling LACP

LACP is disabled by default; you must enable LACP before you begin LACP configuration. You cannot disable LACP while any LACP configuration is present.

LACP learns the capabilities of LAN port groups dynamically and informs the other LAN ports. Once LACP identifies correctly matched Ethernet links, it groups the links into a port channel. The port channel is then added to the spanning tree as a single bridge port.

To configure LACP, you must do the following:

• Enable LACP globally by using the `feature lacp` command.

• You can use different modes for different interfaces within the same LACP-enabled port channel. You can change the mode between `active` and `passive` for an interface only if it is the only interface that is designated to the specified channel group.

SUMMARY STEPS

1. `configure terminal`
2. `feature lacp`
3. `copy running-config startup-config`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><em>Example:</em> <code>switch# configure terminal</code> <code>switch(config)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>feature lacp</code></td>
<td>Enables LACP on the device.</td>
</tr>
<tr>
<td><em>Example:</em> <code>switch(config)# feature lacp</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>copy running-config startup-config</code></td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td><em>Example:</em> <code>switch(config)# copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>
Example

This example shows how to enable LACP:

```
switch# configure terminal
switch (config)# feature lacp
```

Configuring LACP Port-Channel Port Modes

After you enable LACP, you can configure the channel mode for each individual link in the LACP port channel as **active** or **passive**. This channel configuration mode allows the link to operate with LACP.

When you configure port channels with no associated aggregation protocol, all interfaces on both sides of the link remain in the **on** channel mode.

SUMMARY STEPS

1. `configure terminal`
2. `interface type slot/port`
3. `channel-group number mode {active | on | passive}`
4. `show port-channel summary`
5. `copy running-config startup-config`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td></td>
<td>switch (config)#</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>interface type slot/port</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch (config)# interface ethernet 1/4</td>
</tr>
<tr>
<td></td>
<td>switch (config-if)#</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>channel-group number mode {active</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch (config-if)# channel-group 5 mode active</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>show port-channel summary</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch (config-if)# show port-channel summary</td>
</tr>
</tbody>
</table>
Configuring LACP Port-Channel Minimum Links

You can configure the LACP minimum links feature. Although minimum links and max bundles work only in LACP, you can enter the CLI commands for these features for non-LACP port channels, but these commands are nonoperational.

Note

Use the `no lacp min-links` command to restore the default port-channel minimum links configuration.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>no lacp min-links</td>
<td>Restores the default port-channel minimum links configuration.</td>
</tr>
</tbody>
</table>

Example

This example shows how to set the LACP-enabled interface to the active port-channel mode for Ethernet interface 1/4 in channel group 5:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch (config-if)# channel-group 5 mode active
```

### Configuring LACP Port-Channel Minimum Links

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>interface port-channel <em>number</em></td>
<td></td>
</tr>
<tr>
<td>lacp min-links <em>number</em></td>
<td></td>
</tr>
<tr>
<td>show running-config interface port-channel <em>number</em></td>
<td></td>
</tr>
</tbody>
</table>
### Configuring Port Channels

#### Configuring the LACP Port-Channel MaxBundle

You can configure the LACP maxbundle feature. Although minimum links and maxbundles work only in LACP, you can enter the CLI commands for these features for non-LACP port channels, but these commands are nonoperational.

**Note**

Use the `no lacp max-bundle` command to restore the default port-channel max-bundle configuration.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>no lacp max-bundle</code></td>
<td>Restores the default port-channel max-bundle configuration.</td>
</tr>
</tbody>
</table>

**Before you begin**

Ensure that you are in the correct port-channel interface.
SUMMARY STEPS

1. configure terminal
2. interface port-channel number
3. lacp max-bundle number
4. show running-config interface port-channel number

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface port-channel number</td>
<td>Specifies the interface to configure, and enters the interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>switch(config)# interface port-channel 3</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> lacp max-bundle number</td>
<td>Specifies the port-channel interface to configure max-bundle.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Even if the default value is 16, the number of active members in a port channel is the minimum of the pc_max_links_config and pc_max_active_members that is allowed in the port channel.</td>
</tr>
<tr>
<td>switch(config-if)# lacp max-bundle</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> show running-config interface port-channel number</td>
<td>(Optional) Displays the port-channel max-bundle configuration.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>switch(config-if)# show running-config interface port-channel 3</td>
<td></td>
</tr>
</tbody>
</table>

**Example**

This example shows how to configure the port channel interface max-bundle:

```
switch# configure terminal
switch(config)# interface port-channel 3
switch(config-if)# lacp max-bundle 3
```

**Configuring the LACP Fast Timer Rate**

You can change the LACP timer rate to modify the duration of the LACP timeout. Use the `lacp rate` command to set the rate at which LACP control packets are sent to an LACP-supported interface. You can change the
timeout rate from the default rate (30 seconds) to the fast rate (1 second). This command is supported only on LACP-enabled interfaces.

**Note**

We do not recommend changing the LACP timer rate. HA and SSO are not supported when the LACP fast rate timer is configured.

**Note**

Configuring `lacp rate fast` is not recommended on the vPC peer-links. When `lacp rate fast` is configured on the vPC peer-link member interfaces, an alert is displayed in the syslog messages only when the LACP logging level is set to 5.

**Before you begin**

Ensure that you have enabled the LACP feature.

**SUMMARY STEPS**

1. `configure terminal`
2. `interface type slot/port`
3. `lacp rate fast`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure terminal</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td><code>switch# configure terminal</code></td>
</tr>
<tr>
<td></td>
<td><code>switch(config)#</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Specifies the interface to configure and enters the interface configuration mode.</td>
</tr>
<tr>
<td>interface type slot/port</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td><code>switch(config)# interface ethernet 1/4</code></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-if)#</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Configures the fast rate (one second) at which LACP control packets are sent to an LACP-supported interface. To reset the timeout rate to its default, use the <code>no</code> form of the command.</td>
</tr>
<tr>
<td>lacp rate fast</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td><code>switch(config-if)# lacp rate fast</code></td>
</tr>
</tbody>
</table>

**Example**

This example shows how to configure the LACP fast rate on Ethernet interface 1/4:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# lacp rate fast
```
This example shows how to restore the LACP default rate (30 seconds) on Ethernet interface 1/4.

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# no lacp rate fast
```

### Configuring the LACP System Priority

The LACP system ID is the combination of the LACP system priority value and the MAC address.

**Before you begin**

Enable LACP.

### SUMMARY STEPS

1. `configure terminal`
2. `lacp system-priority priority`
3. `show lacp system-identifier`
4. `copy running-config startup-config`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>switch# configure terminal</code> <code>switch(config)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><code>lacp system-priority priority</code></td>
<td>Configures the system priority for use with LACP. Valid values are from 1 through 65535, and higher numbers have a lower priority. The default value is 32768.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>switch(config)# lacp system-priority 40000</code></td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Each VDC has a different LACP system ID because the software adds the MAC address to this configured value.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><code>show lacp system-identifier</code></td>
<td>(Optional) Displays the LACP system identifier.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>switch(config-if)# show lacp system-identifier</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><code>copy running-config startup-config</code></td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>switch(config)# copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>

**Example**

This example shows how to set the LACP system priority to 2500:
Configuring Port Channels

Configuring the LACP Port Priority

When you enable LACP, you can configure each link in the LACP port channel for the port priority.

Before you begin
Enable LACP.

SUMMARY STEPS
1. configure terminal
2. interface type slot/port
3. lacp port-priority priority
4. copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td><strong>Example:</strong> switch# configure terminal switch(config)#</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>interface type slot/port</td>
</tr>
<tr>
<td><strong>Example:</strong> switch(config)# interface ethernet 1/4 switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Specifies the interface that you want to add to a channel group, and enters the interface configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>lacp port-priority priority</td>
</tr>
<tr>
<td><strong>Example:</strong> switch(config-if)# lacp port-priority 40000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Configures the port priority for use with LACP. Valid values are from 1 through 65535, and higher numbers have a lower priority. The default value is 32768.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>copy running-config startup-config</td>
</tr>
<tr>
<td><strong>Example:</strong> switch(config-if)# copy running-config startup-config</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>

Example
This example shows how to set the LACP port priority for Ethernet interface 1/4 to 40000:

switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# lacp port-priority 40000
Configuring LACP System MAC and Role

You can configure the MAC address used by the LACP for protocol exchanges and the optional role. By default, the LACP uses the VDC MAC address. By default, the role is primary.

Use the `no lacp system-mac` command to make LACP use the default (VDC) MAC address and default role.

This procedure is supported on the Cisco Nexus 9336C-FX2, 93300YC-FX2, and 93240YC-FX2-Z switches.

**Before you begin**

LACP must be enabled.

**SUMMARY STEPS**

1. `configure terminal`
2. `lacp system-mac mac-address role role-value`
3. (Optional) `show lacp system-identifier`
4. `copy running-config startup-config`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure terminal</code></td>
<td>Enter global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>switch# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>lacp system-mac mac-address role role-value</code></td>
<td>Specifies the MAC address to use in the LACP protocol exchanges. The role is optional. Primary is the default.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>switch(config)# lacp system-mac 000a.000b.000c role primary</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config)# lacp system-mac 000a.000b.000c role secondary</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>(Optional) <code>show lacp system-identifier</code></td>
<td>Displays the configured MAC address.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>switch(config)# show lacp system-identifier</code></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>copy running-config startup-config</code></td>
<td>Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>switch(config)# copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>

**Example**

The following example shows how to configure the role of a switch as primary.

```
Switch1# sh lacp system-identifier
32768,0-b=0-b=0-b
Switch1# sh run | grep lacp
```
feature lacp
lacp system-mac 000b.000b.000b role primary

The following example shows how to configure the role of a switch as secondary.
Switch2# sh lacp system-identifier
32768,0-b-0-b-0-b
Switch2# sh run | grep lacp
feature lacp
lacp system-mac 000b.000b.000b role secondary

Disabling LACP Graceful Convergence

By default, LACP graceful convergence is enabled. In situations where you need to support LACP interoperability with devices where the graceful failover defaults may delay the time taken for a disabled port to be brought down or cause traffic from the peer to be lost, you can disable convergence. If the downstream access switch is not a Cisco Nexus device, disable the LACP graceful convergence option.

Note
The port channel has to be in the administratively down state before the command can be run.

Before you begin
Enable LACP.

SUMMARY STEPS
1. configure terminal
2. interface port-channel number
3. shutdown
4. no lacp graceful-convergence
5. no shutdown
6. copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>interface port-channel number</td>
<td>Specifies the port channel interface to configure and enters the interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)# interface port-channel 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>shutdown</td>
<td>Administratively shuts down the port channel.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>switch(config-if) shutdown</code></td>
<td>Disables LACP graceful convergence on the port channel.</td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>no lacp graceful-convergence</code></td>
<td>Brings the port channel administratively up.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>switch(config-if)# no lacp graceful-convergence</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>no shutdown</code></td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>switch(config-if) no shutdown</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> <code>copy running-config startup-config</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> <code>switch(config)# copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>

#### Example

This example shows how to disable LACP graceful convergence on a port channel:

```
switch# configure terminal
switch (config)# interface port-channel 1
switch(config-if)# shutdown
switch(config-if)# no lacp graceful-convergence
switch(config-if)# no shutdown
```

### Reenabling LACP Graceful Convergence

If the default LACP graceful convergence is once again required, you can reenable convergence.

**SUMMARY STEPS**

1. `configure terminal`
2. `interface port-channel number`
3. `shutdown`
4. `lacp graceful-convergence`
5. `no shutdown`
6. `copy running-config startup-config`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>switch(config)# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config)#</code></td>
<td></td>
</tr>
</tbody>
</table>
### Configuring Port Channels

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong> interface port-channel number</td>
<td>Specifies the port channel interface to configure and enters the interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# interface port-channel 1</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> shutdown</td>
<td>Administratively shuts down the port channel.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-if) shutdown</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> lACP graceful-convergence</td>
<td>Enables LACP graceful convergence on the port channel.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)# lACP graceful-convergence</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> no shutdown</td>
<td>Brings the port channel administratively up.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-if) no shutdown</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> copy running-config startup-config</td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# copy running-config startup-config</td>
<td></td>
</tr>
</tbody>
</table>

**Example**

This example shows how to enable LACP graceful convergence on a port channel:

```
switch# configure terminal
switch (config)# interface port-channel 1
switch(config-if)# shutdown
switch(config-if)# lACP graceful-convergence
switch(config-if)# no shutdown
```

**Disabling LACP Suspend Individual**

LACP sets a port to the suspended state if it does not receive an LACP PDU from the peer. This process can cause some servers to fail to boot up as they require LACP to logically bring up the port.

**Note**

You should only enter the `lACP suspend-individual` command on edge ports. The port channel has to be in the administratively down state before you can use this command.

**Before you begin**

Enable LACP.
### SUMMARY STEPS

1. `configure terminal`
2. `interface port-channel number`
3. `shutdown`
4. `no lacp suspend-individual`
5. `no shutdown`
6. `copy running-config startup-config`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>switch# configure terminal</code> <code>switch(config)#</code></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>interface port-channel number</code></td>
<td>Specifies the port channel interface to configure and enters the interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>switch(config)# interface port-channel 1</code> <code>switch(config-if)#</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>shutdown</code></td>
<td>Administratively shuts down the port channel.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>switch(config-if) shutdown</code></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>no lacp suspend-individual</code></td>
<td>Disables LACP individual port suspension behavior on the port channel.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>switch(config-if)# no lacp suspend-individual</code></td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td><code>no shutdown</code></td>
<td>Brings the port channel administratively up.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>switch(config-if) no shutdown</code></td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td><code>copy running-config startup-config</code></td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>switch(config)# copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>

### Example

This example shows how to disable LACP individual port suspension on a port channel:

```
switch# configure terminal
switch (config)# interface port-channel 1
switch(config-if)# shutdown
switch(config-if)# no lacp suspend-individual
switch(config-if)# no shutdown
```
Reenabling LACP Suspend Individual

You can reenable the default LACP individual port suspension.

SUMMARY STEPS

1. `configure terminal`
2. `interface port-channel number`
3. `shutdown`
4. `lacp suspend-individual`
5. `no shutdown`
6. `copy running-config startup-config`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure terminal</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Enters global configuration mode.</strong></td>
</tr>
<tr>
<td><code>switch# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>interface port-channel number</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Specifies the port channel interface to configure and enters the interface configuration mode.</strong></td>
</tr>
<tr>
<td><code>switch(config)# interface port-channel 1</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config-if)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>shutdown</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Administratively shuts down the port channel.</strong></td>
</tr>
<tr>
<td><code>switch(config-if) shutdown</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>lacp suspend-individual</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Enables LACP individual port suspension behavior on the port channel.</strong></td>
</tr>
<tr>
<td><code>switch(config-if)# lacp suspend-individual</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>no shutdown</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Brings the port channel administratively up.</strong></td>
</tr>
<tr>
<td><code>switch(config-if) no shutdown</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>copy running-config startup-config</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>(Optional) Copies the running configuration to the startup configuration.</strong></td>
</tr>
<tr>
<td><code>switch(config)# copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>

Example

This example shows how to reenable the LACP individual port suspension on a port channel:
Configuring Delayed LACP

You configure the delayed LACP with the `lacp mode delay` command followed by configuring the LACP port priority.

### Note
For vPC, you must enable the delayed LACP on both vPC switches.

### Note
Delayed LACP is not supported on Layer 3 port channels, FEX modules, or the Cisco Nexus 9500 Series switch.

### Note
For vPC, when the delayed LACP port is on the primary switch and the primary switch fails to boot, you need to remove the vPC configuration on the delayed LACP port-channel of the acting primary switch and flap the port-channel for a new port to be chosen as the delayed LACP port on the existing port-channel.

### Note
When `no lacp suspend-individual` and the delayed LACP feature are configured on the same port, the non-delayed LACP ports belonging to the port are in individual state. When LACP is established, the member should be moved to up state.

As a best practice, do not use `no lacp suspend-individual` together with the delayed LACP feature on the same port channel.

### SUMMARY STEPS
1. configure terminal
2. interface port-channel number
3. lacp mode delay

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>interface port-channel number</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>lacp mode delay</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>To disable delayed LACP, use the <code>no lacp mode delay</code> command.</td>
</tr>
</tbody>
</table>

Complete the configuration of the delayed LACP by configuring the LACP port priority. See the "Configuring the LACP Port Priority" section for details.

The priority of a LACP port determines the election of the delayed LACP port. The port with the lowest numerical priority is elected.

When two or more ports have the same best priority, the VDC system MAC is used to determine which vPC is used. Then within a non-vPC switch or the elected vPC switch, the smallest of the ethernet port names is used.

When the delayed LACP feature is configured and made effective with a port channel flap, the delayed LACP port operates as a member of a regular port channel, allowing data to be exchanged between the server and switch. After receiving the first LACP PDU, the delayed LACP port transitions from a regular port member to a LACP port member.

**Note** The election of the delayed LACP port is not complete or effective until the port channel flaps on the switch or at a remote server.

---

**Example**

The following example configures delayed LACP.

```
switch# config terminal
switch(config)# interface po 1
switch(config-if)# lacp mode delay

switch# config terminal
switch(config)# interface ethernet 1/1
switch(config-if)# lacp port-priority 1
switch(config-if)# channel-group 1 mode active
```

The following example disables delayed LACP.

```
switch# config terminal
switch(config)# interface po 1
switch(config-if)# no lacp mode delay
```
Configuring Port Channel Hash Distribution

Cisco NX-OS supports the adaptive and fixed hash distribution configuration for both global and port-channel levels. This option minimizes traffic disruption by minimizing Result Bundle Hash (RBH) distribution changes when members come up or go down so that flows that are mapped to unchange RBH values continue to flow through the same links. The port-channel level configuration overrules the global configuration. The default configuration is adaptive globally, and there is no configuration for each port channel, so there is no change during an ISSU. No ports are flapped when the command is applied, and the configuration takes effect at the next member link change event. Both modes work with RBH module or non-module schemes.

During an ISSD to a lower version that does not support this feature, you must disable this feature if the fixed mode command is being used globally or if there is a port-channel level configuration.

Configuring Port Channel Hash Distribution at the Global Level

SUMMARY STEPS

1. configure terminal
2. no port-channel hash-distribution {adaptive | fixed}
3. copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td>Step 2 no port-channel hash-distribution {adaptive</td>
<td>fixed}</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# port-channel hash-distribution adaptive</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td>Step 3 copy running-config startup-config</td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# copy running-config startup-config</td>
<td></td>
</tr>
</tbody>
</table>

Example

This example shows how to configure hash distribution at the global level:

```
switch# configure terminal
switch(config)# no port-channel hash-distribution fixed
```
Configuring Port Channel Hash Distribution at the Port Channel Level

SUMMARY STEPS

1. `configure terminal`
2. `interface port-channel {channel-number | range}
3. `no port-channel port hash-distribution {adaptive | fixed}`
4. `copy running-config startup-config`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>switch# configure terminal switch(config)#</code></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>`interface port-channel {channel-number</td>
<td>range}`</td>
</tr>
<tr>
<td></td>
<td>Example: <code>switch# interface port-channel 4 switch(config-if)#</code></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>`no port-channel port hash-distribution {adaptive</td>
<td>fixed}`</td>
</tr>
<tr>
<td></td>
<td>Example: <code>switch(config-if)# port-channel port hash-distribution adaptive switch(config-if)&quot;</code></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><code>copy running-config startup-config</code></td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>switch(config)# copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>

Example

This example shows how to configure hash distribution as a global-level command:

```
switch# configure terminal
switch(config)# no port-channel hash-distribution fixed
```

Enabling ECMP Resilient Hashing

Resilient ECMP ensures minimal impact to the existing flows when members are deleted from an ECMP group. This is achieved by replicating the existing members in a round-robin fashion at the indices that were previously occupied by the deleted members.
### Disable ECMP Resilient Hashing

**Before you begin**

ECMP resilient hashing is enabled.

#### SUMMARY STEPS

1. configure terminal
2. no hardware profile ecmp resilient
3. copy running-config startup-config
4. reload

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> hardware profile ecmp resilient</td>
<td>Enables ECMP resilient hashing and displays the following:</td>
</tr>
<tr>
<td>Example:</td>
<td>Warning: The command will take effect after next reload.</td>
</tr>
<tr>
<td>switch(config)# hardware profile ecmp resilient</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> copy running-config startup-config</td>
<td>Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# copy running-config startup-config</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> reload</td>
<td>Reboots the switch.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# reload</td>
<td></td>
</tr>
</tbody>
</table>
## Configuring Port Channels

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>no hardware profile ecmp resilient</td>
<td>Disables ECMP resilient hashing and displays the following:</td>
</tr>
<tr>
<td>Example:</td>
<td>Warning: The command will take effect after next reload.</td>
</tr>
<tr>
<td>switch(config)# no hardware profile ecmp resilient</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td>copy running-config startup-config</td>
<td>Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# copy running-config startup-config</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td>reload</td>
<td>Reboots the switch.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# reload</td>
<td></td>
</tr>
</tbody>
</table>

### Verifying the ECMP Resilient Hashing Configuration

To display ECMP Resilient Hashing configuration information, perform one of the following tasks:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch(config)# show running-config</td>
<td>grep &quot;hardware profile ecmp resilient&quot; hardware profile ecmp resilient switch(config)#</td>
</tr>
<tr>
<td>switch(config)# show running-config</td>
<td>grep &quot;hardware profile ecmp resilient&quot; switch(config)#</td>
</tr>
</tbody>
</table>

### Verifying the Port-Channel Configuration

To display port-channel configuration information, perform one of the following tasks:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>show interface port-channel channel-number</td>
<td>Displays the status of a port-channel interface.</td>
</tr>
<tr>
<td>show feature</td>
<td>Displays enabled features.</td>
</tr>
<tr>
<td>load-interval {interval seconds [1</td>
<td>2</td>
</tr>
<tr>
<td>show port-channel compatibility-parameters</td>
<td>Displays the parameters that must be the same among the member ports in order to join a port channel.</td>
</tr>
<tr>
<td>show port-channel database [interface port-channel channel-number]</td>
<td>Displays the aggregation state for one or more port-channel interfaces.</td>
</tr>
</tbody>
</table>
### Monitoring the Port-Channel Interface Configuration

Use the following commands to display port-channel interface configuration information.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>clear counters interface port-channel</td>
<td>Clears the counters.</td>
</tr>
<tr>
<td>clear lacp counters [interface port-channel</td>
<td>Clears the LACP counters.</td>
</tr>
<tr>
<td>channel-number]</td>
<td></td>
</tr>
<tr>
<td>load- interval {interval seconds {1</td>
<td>2</td>
</tr>
<tr>
<td>show interface counters [module module]</td>
<td>Displays input and output octets unicast packets, multicast packets, and broadcast packets.</td>
</tr>
<tr>
<td>show interface counters detailed [all]</td>
<td>Displays input packets, bytes, and multicast and output packets and bytes.</td>
</tr>
<tr>
<td>show interface counters errors [module module]</td>
<td>Displays information about the number of error packets.</td>
</tr>
<tr>
<td>show lacp counters</td>
<td>Displays statistics for LACP.</td>
</tr>
</tbody>
</table>

### Example Configurations for Port Channels

This example shows how to create an LACP port channel and add two Layer 2 interfaces to that port channel:
switch# configure terminal
switch (config)# feature lacp
switch (config)# interface port-channel 5
switch (config-if)# interface ethernet 1/4
switch(config-if)# switchport
switch(config-if)# channel-group 5 mode active
switch(config-if)# lacp port priority 40000
switch(config-if)# interface ethernet 1/7
switch(config-if)# switchport
switch(config-if)# channel-group 5 mode

This example shows how to add two Layer 3 interfaces to a channel group. The Cisco NX-OS software automatically creates the port channel:

switch# configure terminal
switch (config)# interface ethernet 1/5
switch(config-if)# no switchport
switch(config-if)# no ip address
switch(config-if)# channel-group 6 mode active
switch (config)# interface ethernet 2/5
switch(config-if)# no switchport
switch(config-if)# no ip address
switch(config-if)# channel-group 6 mode active
switch (config)# interface port-channel 6
switch(config-if)# ip address 192.0.2.1/8

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>System management</td>
<td>Cisco Nexus 9000 Series NX-OS System Management Configuration Guide</td>
</tr>
<tr>
<td>High availability</td>
<td>Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide</td>
</tr>
<tr>
<td>Licensing</td>
<td>Cisco NX-OS Licensing Guide</td>
</tr>
</tbody>
</table>
CHAPTER 8

Configuring vPCs

- Information About vPCs, on page 219
- Licensing Requirements for vPCs, on page 250
- Guidelines and Limitations, on page 251
- Default Settings, on page 253
- Configuring vPCs, on page 254
- Verifying the vPC Configuration, on page 290
- Monitoring vPCs, on page 290
- Configuration Examples for vPCs, on page 290
- Related Documents, on page 293

Information About vPCs

vPC Overview

A virtual port channel (vPC) allows links that are physically connected to two different Cisco Nexus 9000 Series devices to appear as a single port channel by a third device (see figure). The third device can be a switch, server, or any other networking device that supports port channels. A vPC can provide Layer 2 multipathing, which allows you to create redundancy and increase the bisectional bandwidth by enabling multiple parallel paths between nodes and allowing load balancing traffic.

Figure 11: vPC Architecture
You can use only Layer 2 port channels in the vPC. You configure the port channels by using one of the following:

- No protocol
- Link Aggregation Control Protocol (LACP)

When you configure the port channels in a vPC—including the vPC peer link channel—without using LACP, each device can have up to eight active links in a single port channel. When you configure the port channels in a vPC—including the vPC peer link channels—using LACP, each device can have eight active links and eight standby links in a single port channel. (See the “vPC Interactions with Other Features” section for more information on using LACP and vPCs.)

You must enable the vPC feature before you can configure or run the vPC functionality.

The system automatically takes a checkpoint prior to disabling the feature, and you can roll back to this checkpoint.

After you enable the vPC functionality, you create the peer-keepalive link, which sends heartbeat messages between the two vPC peer devices.

You can create a vPC peer link by configuring a port channel on one Cisco Nexus 9000 Series chassis by using two or more 10-Gigabit Ethernet ports or 40-Gigabit Ethernet ports. To ensure that you have the correct hardware to enable and run a vPC, enter the `show hardware feature-capability` command. If you see an X across from the vPC in your command output, your hardware cannot enable the vPC feature.

We recommend that you configure the vPC peer link Layer 2 port channels as trunks. On another Cisco Nexus 9000 Series chassis, you configure another port channel again using two or more 10-Gigabit Ethernet ports or 40-Gigabit Ethernet ports in the dedicated port mode. Connecting these two port channels creates a vPC peer link in which the two linked Cisco Nexus devices appear as one device to a third device. The third device, or downstream device, can be a switch, server, or any other networking device that uses a regular port channel to connect to the vPC. If you are not using the correct module, the system displays an error message.

We recommend that you configure the vPC peer links on dedicated ports of different modules to reduce the possibility of a failure. For the best resiliency scenario, use at least two modules.

If you must configure all the vPC peer links and core-facing interfaces on a single module, you should configure a track object that is associated with the Layer 3 link to the core and on all the links on the vPC peer link on both vPC peer devices. Once you configure this feature and if the primary vPC peer device fails, the system automatically suspends all the vPC links on the primary vPC peer device. This action forces all the vPC traffic to the secondary vPC peer device until the system stabilizes.

You can create a track object and apply that object to all links on the primary vPC peer device that connect to the core and to the vPC peer link. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for information about the `track interface` command.

The vPC domain includes both vPC peer devices, the vPC peer-keepalive link, the vPC peer link, and all of the port channels in the vPC domain connected to the downstream device. You can have only one vPC domain ID on each device.

In this version, you can connect each downstream device to a single vPC domain ID using a single port channel.
Always attach all vPC devices using port channels to both vPC peer devices.

A vPC (see figure) provides the following benefits:

- Allows a single device to use a port channel across two upstream devices
- Eliminates Spanning Tree Protocol (STP) blocked ports
- Provides a loop-free topology
- Uses all available uplink bandwidth
- Provides fast convergence if either the link or a device fails
- Provides link-level resiliency
- Assures high availability

**Figure 12: vPC Interfaces**

**Hitless vPC Role Change**

A virtual port channel (vPC) allows links that are physically connected to two different Cisco Nexus 9000 Series devices to appear as a single port channel. The vPC role change feature enables you switch vPC roles between vPC peers without impacting traffic flow. The vPC role switching is done based on the role priority value of the device under the vPC domain. A vPC peer device with lower role priority is selected as the primary vPC device during the vPC Role switch. You can use the vpc role preempt command to switch vPC role between peers.

For information about how to configure Hitless vPC Role Change, see Configuring Hitless vPC Role Change, on page 287.
vPC Terminology

The terminology used in vPCs is as follows:

- **vPC**—The combined port channel between the vPC peer devices and the downstream device.

- **vPC peer device**—One of a pair of devices that are connected with the special port channel known as the vPC peer link.

- **vPC peer link**—The link used to synchronize states between the vPC peer devices. Both ends must be on 10-Gigabit Ethernet or 40-Gigabit Ethernet interfaces.

- **vPC member port**—An interface that belongs to a vPC.

- **Host vPC port**—A Fabric Extender host interfaces that belongs to a vPC.

- **vPC domain**—This domain includes both vPC peer devices, the vPC peer-keepalive link, and all of the port channels in the vPC connected to the downstream devices. It is also associated to the configuration mode that you must use to assign vPC global parameters.

- **vPC peer-keepalive link**—The peer-keepalive link monitors the vitality of a vPC peer Cisco Nexus 9000 Series device. The peer-keepalive link sends configurable, periodic keepalive messages between vPC peer devices.

We recommend that you associate a peer-keepalive link to a separate virtual routing and forwarding (VRF) instance that is mapped to a Layer 3 interface in each vPC peer device. If you do not configure a separate VRF, the system uses the management VRF by default. However, if you use the management interfaces for the peer-keepalive link, you must put a management switch connected to both the active and standby management ports on each vPC peer device (see figure).
Figure 13: Separate Switch Required to Connect Management Ports for vPC Peer-Keepalive Link

No data or synchronization traffic moves over the vPC peer-keepalive link; the only traffic on this link is a message that indicates that the originating switch is operating and running a vPC.

- **vPC member port** — Interfaces that belong to the vPCs.
- **Dual-active** — Both vPC peers act as primary. This situation occurs when the peer-keepalive and peer-link go down when both the peers are still active. In this case, the secondary vPC assumes that the primary vPC is inactive and acts as the primary vPC.
- **Recovery** — When the peer-keepalive and the peer-link come up, one switch becomes the secondary vPC. On the switch that becomes the secondary vPC, the vPC links go down and come back up.

**vPC Peer Link Overview**

You can have only two devices as vPC peers; each device can serve as a vPC peer to only one other vPC peer. The vPC peer devices can also have non-vPC links to other devices.

See the following figure for invalid vPC peer configurations.
To make a valid configuration, you first configure a port channel on each device and then configure the vPC domain. You assign the port channel on each device as a peer link, using the same vPC domain ID. For redundancy, we recommend that you should configure at least two of the dedicated ports into the port channel because if one of the interfaces in the vPC peer link fails, the device automatically falls back to use another interface in the peer link.

**Note**

Many operational parameters and configuration parameters must be the same in each device connected by a vPC peer link (see the “Compatibility Parameters for vPC Interfaces” section). Because each device is completely independent on the management plane, you must ensure that the devices are compatible on the critical parameters. vPC peer devices have separate control planes. After configuring the vPC peer link, you should display the configuration on each vPC peer device to ensure that the configurations are compatible.

**Note**

You must ensure that the two devices connected by the vPC peer link have certain identical operational and configuration parameters. For more information on required configuration consistency, see the “Compatibility Parameters for vPC Interfaces” section.

When you configure the vPC peer link, the vPC peer devices negotiate that one of the connected devices is the primary device and the other connected device is the secondary device (see the “Configuring vPCs” section). The Cisco NX-OS software uses the lowest MAC address to elect the primary device. The software takes different actions on each device—that is, the primary and secondary—only in certain failover conditions. If the primary device fails, the secondary device becomes the new primary device when the system recovers, and the previously primary device is now the secondary device.

You can also configure which of the vPC devices is the primary device. Changing the priority of the vPC peer devices can cause the interfaces in your network to go up and down. If you want to configure the role priority again to make one vPC device the primary device, configure the role priority on both the primary vPC device with a lower priority value and the secondary vPC device with the higher value. Then, shut down the port channel that is the vPC peer link on both devices by entering the **shutdown** command, and finally reenable the port channel on both devices by entering the **no shutdown** command.
We recommend that you use two different modules for redundancy on each vPC peer device on each vPC peer link.

The software keeps all traffic that forwards across the vPC peer devices as local traffic. A packet that ingresses the port channel uses one of the local links rather than moving across the vPC peer link. Unknown unicast, multicast, and broadcast traffic (including STP BPDUs) are flooded across the vPC peer link. The software keeps the multicast forwarding state synchronized on both of the vPC peer devices.

You can configure any of the standard load-balancing schemes on both the vPC peer link devices and the downstream device (see the “Configuring Port Channels” chapter for information about load balancing).

Configuration information flows across the vPC peer links using the Cisco Fabric Services over Ethernet (CFSoE) protocol. (See the “vPC and Orphan Ports” section for more information about CFSoE.)

All MAC addresses for those VLANs configured on both devices are synchronized between vPC peer devices. The software uses CFSoE for this synchronization. (See the “vPC and Orphan Ports” section for information about CFSoE.)

If the vPC peer link fails, the software checks the status of the remote vPC peer device using the peer-keepalive link, which is a link between vPC peer devices that ensures that both devices are up. If the vPC peer device is up, the secondary vPC device disables all vPC ports on its device, to prevent loops and disappearing or flooding traffic. The data then forwards down the remaining active links of the port channel.

We recommend that you create and configure a separate VRF and configure a Layer 3 port on each vPC peer device in that VRF for the vPC peer-keepalive link. The default ports and VRF for the peer-keepalive are the management ports and VRF.

The software learns of a vPC peer device failure when the keepalive messages are not returned over the peer-keepalive link.

Use a separate link (vPC peer-keepalive link) to send configurable keepalive messages between the vPC peer devices. The keepalive messages on the vPC peer-keepalive link determines whether a failure is on the vPC peer link only or on the vPC peer device. The keepalive messages are used only when all the links in the peer link fail. See the “Peer-Keepalive Link and Messages” section for information about the keepalive message.

**Features That You Must Manually Configure on the Primary and Secondary Devices**

You must manually configure the following features to conform to the primary/secondary mapping of each of the vPC peer devices:

- **STP root**—Configure the primary vPC peer device as the STP primary root device and configure the vPC secondary device to be the STP secondary root device. See the “vPC Peer Links and STP” section for more information about vPCs and STP.
  - We recommend that you configure the vPC peer link interfaces as STP network ports so that Bridge Assurance is enabled on all vPC peer links.
  - We recommend that you configure Rapid per VLAN Spanning Tree plus (PVST+) so that the primary device is the root for all VLANs and configure Multiple Spanning Tree (MST) so that the primary device is the root for all instances.
• Layer 3 VLAN network interface—Configure Layer 3 connectivity from each vPC peer device by configuring a VLAN network interface for the same VLAN from both devices.

• HSRP active—If you want to use Hot Standby Router Protocol (HSRP) and VLAN interfaces on the vPC peer devices, configure the primary vPC peer device with the HSRP active highest priority. Configure the secondary device to be the HSRP standby and ensure that you have VLAN interfaces on each vPC device that are in the same administrative and operational mode. (See the “vPC Peer Links and Routing” section for more information on vPC and HSRP.)

While you configure Unidirectional Link Detection (UDLD), note the following recommendations:

• If LACP is used as port-channel aggregation protocol, UDLD is not required in a vPC domain.

• If LACP is not used as the port-channel aggregation protocol (static port-channel), use UDLD in normal mode on vPC member ports.

• If STP is used without Bridge Assurance and if LACP is not used, use UDLD in normal mode on vPC orphan ports.

### Configuring Layer 3 Backup Routes on a vPC Peer Link

You can use VLAN network interfaces on the vPC peer devices to link to Layer 3 of the network for such applications as HSRP and PIM. Ensure that you have a VLAN network interface configured on each peer device and that the interface is connected to the same VLAN on each device. Also, each VLAN interface must be in the same administrative and operational mode. For more information about configuring VLAN network interfaces, see the “Configuring Layer 3 Interfaces” chapter.

If a failover occurs on the vPC peer link, the VLAN interfaces on the vPC peer devices are also affected. If a vPC peer link fails, the system brings down associated VLAN interfaces on the secondary vPC peer device. You can ensure that specified VLAN interfaces do not go down on the vPC secondary device when the vPC peer link fails.

Use the `dual-active exclude interface-vlan` command to configure this feature.

### Peer-Keepalive Link and Messages

The Cisco NX-OS software uses the peer-keepalive link between the vPC peers to transmit periodic, configurable keepalive messages. You must have Layer 3 connectivity between the peer devices to transmit these messages; the system cannot bring up the vPC peer link unless the peer-keepalive link is already up and running.

---

**Note**

We recommend that you associate the vPC peer-keepalive link to a separate VRF mapped to a Layer 3 interface in each vPC peer device. If you do not configure a separate VRF, the system uses the management VRF and management ports by default. Do not use the peer link itself to send and receive vPC peer-keepalive messages.

---

If one of the vPC peer devices fails, the vPC peer device on the other side of the vPC peer link senses the failure by not receiving any peer-keepalive messages. The default interval time for the vPC peer-keepalive message is 1 second, and you can configure the interval between 400 milliseconds and 10 seconds.

You can configure a hold-timeout value with a range of 3 to 10 seconds; the default hold-timeout value is 3 seconds. This timer starts when the vPC peer link goes down. During this hold-timeout period, the secondary
vPC peer device ignores vPC peer-keepalive messages, which ensures that network convergence occurs before a vPC action takes place. The purpose of the hold-timeout period is to prevent false-positive cases.

You can also configure a timeout value with a range of 3 to 20 seconds; the default timeout value is 5 seconds. This timer starts at the end of the hold-timeout interval. During the timeout period, the secondary vPC peer device checks for vPC peer-keepalive hello messages from the primary vPC peer device. If the secondary vPC peer device receives a single hello message, that device disables all vPC interfaces on the secondary vPC peer device.

The difference between the hold-timeout and the timeout parameters is as follows:

- During the hold-timeout, the vPC secondary device does not take any action based on any keepalive messages received, which prevents the system taking action when the keepalive might be received just temporarily, such as if a supervisor fails a few seconds after the peer link goes down.
- During the timeout, the vPC secondary device takes action to become the vPC primary device if no keepalive message is received by the end of the configured interval.

See the “Configuring vPCs” section for information about configuring the timer for the keepalive messages.

---

**Note**

Ensure that both the source and destination IP addresses used for the peer-keepalive messages are unique in your network and these IP addresses are reachable from the VRF associated with the vPC peer-keepalive link.

Peer-keepalive IP addresses must be global unicast addresses. Link-local addresses are not supported.

Use the command-line interface (CLI) to configure the interfaces you are using the vPC peer-keepalive messages as trusted ports. Leave the precedence at the default (6) or configure it higher. The following is an example of configuring an interface as a trusted port:

```
(config)# class-map type qos match-all trust-map
(config-cmap-qos)# match cos 4-7
(config)# policy-map type qos ingrespolcy
(config-pmap-qos)# class trust-map
(config)# interface Ethernet 8/11
(config-if)# service-policy type qos input ingrespolcy
```

---

**vPC Peer-Gateway**

You can configure vPC peer devices to act as the gateway even for packets that are destined to the vPC peer device’s MAC address.

Use the `peer-gateway` command to configure this feature.

---

**Note**

The `peer-gateway exclude-vlan` command that is used when configuring a VLAN interface for Layer 3 backup routing on vPC peer devices is not supported.

Some network-attached storage (NAS) devices or load balancers might have features that help to optimize the performances of particular applications. These features enable the device to avoid a routing-table lookup when responding to a request that originated from a host that is not locally attached to the same subnet. Such devices might reply to traffic using the MAC address of the sender Cisco Nexus 9000 Series device rather
than the common HSRP gateway. This behavior is noncompliant with some basic Ethernet RFC standards. Packets that reach a vPC device for the nonlocal router MAC address are sent across the peer link and could be dropped by the built in vPC loop avoidance mechanism if the final destination is behind another vPC.

The vPC peer-gateway capability allows a vPC switch to act as the active gateway for packets that are addressed to the router MAC address of the vPC peer. This feature enables local forwarding of packets without the need to cross the vPC peer link. In this scenario, the feature optimizes use of the peer link and avoids potential traffic loss.

Configuring the peer-gateway feature must be done on both primary and secondary vPC peers and is nondisruptive to the operations of the device or to the vPC traffic. The vPC peer-gateway feature can be configured globally under the vPC domain submode.

When you enable this feature, Cisco NX-OS automatically disables IP redirects on all interface VLANs mapped over a vPC VLAN to avoid generation of IP redirect messages for packets switched through the peer gateway router.

Packets that arrive at the peer-gateway vPC device have their Time to Live (TTL) decremented, so that packets carrying a TTL of 1 might get dropped in transit due to TTL expiration. You should take this situation into account when the peer-gateway feature is enabled and particular network protocols that source packets with a TTL of 1 operate on a vPC VLAN.

**vPC Domain**

You can use the vPC domain ID to identify the vPC peer links and the ports that are connected to the vPC downstream devices.

The vPC domain is also a configuration mode that you use to configure the keepalive messages and other vPC peer link parameters rather than accept the default values. See the “Configuring vPCs” section for more information about configuring these parameters.

To create a vPC domain, you must first create a vPC domain ID on each vPC peer device using a number from 1 to 1000. You can have only one vPC domain per vPC peer.

You must explicitly configure the port channel that you want to act as the peer link on each device. You associate the port channel that you made a peer link on each device with the same vPC domain ID to form a single vPC domain. Within this domain, the system provides a loop-free topology and Layer 2 multipathing.

You can only configure these port channels and vPC peer links statically. All ports in the vPC on each of the vPC peer devices must be in the same VDC. You can configure the port channels and vPC peer links either using LACP or no protocol. We recommend that you use LACP with the interfaces in active mode to configure port channels in each vPC, which ensures an optimized, graceful recovery in a port-channel failover scenario and provides configuration checks against configuration mismatches among the port channels themselves.

The vPC peer devices use the vPC domain ID that you configure to automatically assign a unique vPC system MAC address. Each vPC domain has a unique MAC address that is used as a unique identifier for the specific vPC-related operations, although the devices use the vPC system MAC addresses only for link-scope operations, such as LACP. We recommend that you create each vPC domain within the contiguous Layer 2 network with a unique domain ID. You can also configure a specific MAC address for the vPC domain, rather than having the Cisco NX-OS software assign the address.

See the “vPC and Orphan Ports” section for more information about displaying the vPC MAC table.

After you create a vPC domain, the Cisco NX-OS software creates a system priority for the vPC domain. You can also configure a specific system priority for the vPC domain.
When manually configuring the system priority, you must ensure that you assign the same priority value on both vPC peer devices. If the vPC peer devices have different system priority values, vPC does not come up.

vPC Topology

The following figure shows a basic configuration in which the Cisco Nexus 9000 Series device ports are directly connected to another switch or host and are configured as part of a port channel that becomes part of a vPC.

*Figure 15: Switch vPC Topology*

In the figure, vPC 20 is configured on port channel 20, which has Eth1/10 on the first device and Eth2/1 on the second as member ports.

You can configure a vPC from the peer devices through Fabric Extenders (FEXs) as shown in the figure.
In the figure, each FEX is single-homed (straight-through FEX topology) with a Cisco Nexus 9000 Series device. The host interfaces on this FEX are configured as port channels and those port channels are configured as vPCs. Eth101/1/1 and Eth102/1/5 are configured as members of PO200, and PO200 is configured for vPC 200.

In both topologies, port channels P020 and P0200 must be configured identically on the peer switches and configuration synchronization is used to synchronize the configurations of the vPC switches.


**Compatibility Parameters for vPC Interfaces**

Many configuration and operational parameters must be identical on all interfaces in the vPC. We recommend that you configure the Layer 2 port channels that you use for the vPC peer link in trunk mode.

After you enable the vPC feature and configure the peer link on both vPC peer devices, Cisco Fabric Services (CFS) messages provide a copy of the configuration on the local vPC peer device configuration to the remote vPC peer device. The system then determines whether any of the crucial configuration parameters differ on the two devices. (See the “vPC and Orphan Ports” section for more information about CFS.)

**Note**

Enter the `show vpc consistency-parameters` command to display the configured values on all interfaces in the vPC. The displayed configurations are only those configurations that would limit the vPC peer link and vPC from coming up.

The compatibility check process for vPCs differs from the compatibility check for regular port channels.
See the “Configuring Port Channels” chapter for information about regular port channels.

**Configuration Parameters That Must Be Identical**

The configuration parameters in this section must be configured identically on both devices of the vPC peer link; otherwise, the vPC moves fully or partially into a suspended mode.

---

**Note**

You must ensure that all interfaces in the vPC have the identical operational and configuration parameters listed in this section.

---

**Note**

Enter the `show vpc consistency-parameters` command to display the configured values on all interfaces in the vPC. The displayed configurations are only those configurations that would limit the vPC peer link and vPC from coming up.

---

The devices automatically check for compatibility for some of these parameters on the vPC interfaces. The per-interface parameters must be consistent per interface, and the global parameters must be consistent globally:

- Port-channel mode: on, off, or active (port-channel mode can, however, be active/passive on each side of the vPC peer)
- Link speed per channel
- Duplex mode per channel
- Trunk mode per channel:
  - Native VLAN
  - VLANs allowed on trunk
  - Tagging of native VLAN traffic
- Spanning Tree Protocol (STP) mode
- STP region configuration for Multiple Spanning Tree
- Enable/disable state per VLAN
- STP global settings:
  - Bridge Assurance setting
  - Port type setting
  - Loop Guard settings
- STP interface settings:
  - Port type setting
  - Loop Guard
  - Root Guard
• Maximum Transmission Unit (MTU)

If any of these parameters are not enabled or defined on either device, the vPC consistency check ignores those parameters.

---

**Note**

To ensure that none of the vPC interfaces are in the suspend mode, enter the `show vpc brief` and `show vpc consistency-parameters` commands and check the syslog messages.

---

### Configuration Parameters That Should Be Identical

When any of the following parameters are not configured identically on both vPC peer devices, a misconfiguration might cause undesirable behavior in the traffic flow:

• MAC aging timers
• Static MAC entries
• VLAN interface—Each device on the end of the vPC peer link must have a VLAN interface configured for the same VLAN on both ends and they must be in the same administrative and operational mode. Those VLANs configured on only one device of the peer link do not pass traffic using the vPC or peer link. You must create all VLANs on both the primary and secondary vPC devices, or the VLAN will be suspended.
• All ACL configurations and parameters
• Quality of Service (QoS) configuration and parameters
• STP interface settings:
  • BPDU Filter
  • BPDU Guard
  • Cost
  • Link type
  • Priority
  • VLANs (Rapid PVST+)
• Port security
• Cisco Trusted Security (CTS)
• Dynamic Host Configuration Protocol (DHCP) snooping
• Network Access Control (NAC)
• Dynamic ARP Inspection (DAI)
• IP source guard (IPSG)
• Internet Group Management Protocol (IGMP) snooping
• Hot Standby Routing Protocol (HSRP)
- Protocol Independent Multicast (PIM)
- All routing protocol configurations

To ensure that all the configuration parameters are compatible, we recommend that you display the configurations for each vPC peer device once you configure the vPC.

**Consequences of Parameter Mismatches**

You can configure the graceful consistency check feature, which suspends only the links on the secondary peer device when a mismatch is introduced in a working vPC. This feature is configurable only in the CLI and is enabled by default.

Use the graceful consistency-check command to configure this feature.

As part of the consistency check of all parameters from the list of parameters that must be identical, the system checks the consistency of all VLANs.

The vPC remains operational, and only the inconsistent VLANs are brought down. This per-VLAN consistency check feature cannot be disabled and does not apply to Multiple Spanning Tree (MST) VLANs.

**vPC Number**

Once you have created the vPC domain ID and the vPC peer link, you create port channels to attach the downstream device to each vPC peer device. That is, you create one port channel to the downstream device from the primary vPC peer device and you create another port channel to the downstream device from the secondary peer device.

---

**Note**

We recommend that you configure the ports on the downstream devices that connect to a host or a network device that is not functioning as a switch or a bridge as STP edge ports.

On each vPC peer device, you assign a vPC number to the port channel that connects to the downstream device. You will experience minimal traffic disruption when you are creating vPCs. To simplify the configuration, you can assign the vPC ID number to every port channel to be the same as the port channel itself (that is, vPC ID 10 for port channel 10).

---

**Note**

The vPC number that you assign to the port channel that connects to the downstream device from the vPC peer device must be identical on both vPC peer devices.

---

**Moving Other Port Channels into a vPC**

---

**Note**

You must attach a downstream device using a port channel to both vPC peer devices.

To connect to the downstream device, you create a port channel to the downstream device from the primary vPC peer device and you create another port channel to the downstream device from the secondary peer device.
On each vPC peer device, you assign a vPC number to the port channel that connects to the downstream device. You will experience minimal traffic disruption when you are creating vPCs.

### Configuring vPC Peer Links and Links to the Core on a Single Module

We recommend that you configure the vPC peer links on dedicated ports of different modules to reduce the possibility of a failure. For the best resiliency scenario, use at least two modules.

If you must configure all the vPC peer links and core-facing interfaces on a single module, you should configure, using the command-line interface, a track object and a track list that is associated with the Layer 3 link to the core and on all vPC peer links on both vPC peer devices. You use this configuration to avoid dropping traffic if that particular module goes down because when all the tracked objects on the track list go down, the system does the following:

- Stops the vPC primary peer device sending peer-keepalive messages, which forces the vPC secondary peer device to take over.
- Brings down all the downstream vPCs on that vPC peer device, which forces all the traffic to be rerouted in the access switch toward the other vPC peer device.

Once you configure this feature and if the module fails, the system automatically suspends all the vPC links on the primary vPC peer device and stops the peer-keepalive messages. This action forces the vPC secondary device to take over the primary role and all the vPC traffic to go to this new vPC primary device until the system stabilizes.

You should create a track list that contains all the links to the core and all the vPC peer links as its object. Enable tracking for the specified vPC domain for this track list. Apply this same configuration to the other vPC peer device. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for information about configuring object tracking and track lists.

This example uses Boolean OR in the track list and forces all traffic to the vPC peer device only for a complete module failure. If you want to trigger a switchover when any core interface or peer link goes down, use a Boolean AND in the torack list below.

To configure a track list to switch over a vPC to the remote peer when all related interfaces on a single module fail, follow these steps:

1. Configure track objects on an interface (Layer 3 to core) and on a port channel (vPC peer link).

   ```
   switch(config-if)# track 35 interface ethernet 8/35 line-protocol
   switch(config-track)# track 23 interface ethernet 8/33 line-protocol
   switch(config)# track 55 interface port-channel 100 line-protocol
   ```

2. Create a track list that contains all the interfaces in the track list using the Boolean OR to trigger when all objects fail.

   ```
   switch(config)# track 44 list boolean OR
   switch(config-track)# object 23
   ```
3. Add this track object to the vPC domain:

   switch(config)# vpc domain 1
   switch(config-vpc-domain)# track 44

4. Display the track object:

   switch# show vpc brief
   
   Legend:
   (*) - local vPC is down, forwarding via vPC peer-link
   vPC domain id : 1
   Peer status : peer adjacency formed ok
   vPC keep-alive status : peer is alive
   Configuration consistency status: success
   vPC role : secondary
   Number of vPCs configured : 52
   Track object : 44
   vPC Peer-link status
   
<table>
<thead>
<tr>
<th>id</th>
<th>Port</th>
<th>Status</th>
<th>Active vlans</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Po100</td>
<td>up</td>
<td>1-5,140</td>
</tr>
</tbody>
</table>
   
   vPC status
   
<table>
<thead>
<tr>
<th>id</th>
<th>Port</th>
<th>Status</th>
<th>Consistency Reason</th>
<th>Active vlans</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Po100</td>
<td>up</td>
<td>success</td>
<td>1-5,140</td>
</tr>
</tbody>
</table>

   This example shows how to display information about the track objects:

   switch# show track brief
   
   Track Type Instance Parameter State Last Change
   23 Interface Ethernet8/33 Line Protocol UP 00:03:05
   35 Interface Ethernet8/35 Line Protocol UP 00:03:15
   44 List ----- Boolean UP 00:01:19
   55 Interface port-channel100 Line Protocol UP 00:00:34

**vPC Interactions with Other Features**

**vPC and LACP**

LACP uses the system MAC address of the vPC domain to form the LACP Aggregation Group (LAG) ID for the vPC. (See the “Configuring Port Channels” chapter for information about LAG-ID and LACP.) You can use LACP on all the vPC port channels, including those channels from the downstream device. We recommend that you configure LACP with active mode on the interfaces on each port channel on the vPC peer devices. This configuration allows you to more easily detect compatibility between devices, unidirectional links, and multihop connection, and provides dynamic reaction to run-time changes and link failures.
We recommend that you manually configure the system priority on the vPC peer link devices to ensure that the vPC peer link devices have a higher LACP priority than the downstream connected devices. A lower numerical value system priority means a higher LACP priority.

---

**Note**

When manually configuring the system priority, you must ensure that you assign the same priority value on both vPC peer devices. If the vPC peer devices have different system priority values, vPC does not come up.

---

**vPC Peer Links and STP**

Although vPCs provide a loop-free Layer 2 topology, STP is still required to provide a fail-safe mechanism to protect against any incorrect or defective cabling or possible misconfiguration. When you first bring up a vPC, STP reconverges. STP treats the vPC peer link as a special link and always includes the vPC peer link in the STP active topology.

We recommend that you set all the vPC peer link interfaces to the STP network port type so that Bridge Assurance is automatically enabled on all vPC peer links. We also recommend that you do not enable any of the STP enhancement features on vPC peer links. If the STP enhancements are already configured, they do not cause any problems for the vPC peer links.

When you are running both MST and Rapid PVST+, ensure that the PVST simulation feature is correctly configured.

See the *Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide* for information about STP enhancement features and PVST simulation.

---

**Note**

You must configure a list of parameters to be identical on the vPC peer devices on both sides of the vPC peer link. See the “Compatibility Parameters for vPC Interfaces” section for information about these required matched settings.

---

STP is distributed; that is, the protocol continues running on both vPC peer devices. However, the configuration on the vPC peer device elected as the primary device controls the STP process for the vPC interfaces on the secondary vPC peer device.

The primary vPC device synchronizes the STP state on the vPC secondary peer device using Cisco Fabric Services over Ethernet (CFSoE). See the “vPC and Orphan Ports” section for information about CFSoE.

The STP process for vPC also relies on the periodic keepalive messages to determine when one of the connected devices on the peer link fails. See the “Peer-Keepalive Link and Messages” section for information about these messages.

The vPC manager performs a proposal/handshake agreement between the vPC peer devices that set the primary and secondary devices and coordinates the two devices for STP. The primary vPC peer device then controls the STP protocol on both the primary and secondary devices. We recommend that you configure the primary vPC peer device as the STP primary root device and configure the secondary VPC device to be the STP secondary root device.

If the primary vPC peer device fails over to the secondary vPC peer device, there is no change in the STP topology.

The BPDUs uses the MAC address set for the vPC for the STP bridge ID in the designated bridge ID field. The vPC primary device sends these BPDUs on the vPC interfaces.
You must configure both ends of vPC peer link with the identical STP configuration for the following parameters:

- **STP global settings:**
  - STP mode
  - STP region configuration for MST
  - Enable/disable state per VLAN
  - Bridge Assurance setting
  - Port type setting
  - Loop Guard settings

- **STP interface settings:**
  - Port type setting
  - Loop Guard
  - Root Guard

---

**Note**

If any of these parameters are misconfigured, the Cisco NX-OS software suspends all interfaces in the vPC. Check the syslog and enter the `show vpc brief` command to see if the vPC interfaces are suspended.

---

Ensure that the following STP interface configurations are identical on both sides of the vPC peer links or you may see unpredictable behavior in the traffic flow:

- BPDU Filter
- BPDU Guard
- Cost
- Link type
- Priority
- VLANs (PVRST+)

---

**Note**

Display the configuration on both sides of the vPC peer link to ensure that the settings are identical.

---

You can use the `show spanning-tree` command to display information about the vPC when that feature is enabled. See the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for an example.

---

**Note**

We recommend that you configure the ports on the downstream devices as STP edge ports. You should configure all host ports connected to a switch as STP edge ports. See the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for more information about STP port types.
vPC Peer Switch

The vPC peer switch feature was added to Cisco NX-OS to address performance concerns around STP convergence. This feature allows a pair of Cisco Nexus 9000 Series devices to appear as a single STP root in the Layer 2 topology. This feature eliminates the need to pin the STP root to the vPC primary switch and improves vPC convergence if the vPC primary switch fails.

To avoid loops, the vPC peer link is excluded from the STP computation. In vPC peer switch mode, STP BPDUs are sent from both vPC peer devices to avoid issues related to STP BPDU timeout on the downstream switches, which can cause traffic disruption.

This feature can be used with the pure peer switch topology in which the devices all belong to the vPC.

---

**Note**

Peer-switch feature is supported on networks that use vPC and STP-based redundancy is not supported. If the vPC peer-link fail in a hybrid peer-switch configuration, you can lose traffic. In this scenario, the vPC peers use the same STP root ID as well as the same bridge ID. The access switch traffic is split in two with half going to the first vPC peer and the other half to the second vPC peer. With peer link failure, there is no impact to the north/south traffic but the east/west traffic is lost.

See the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide for information about STP enhancement features and Rapid PVST+.

vPC and ARP or ND

A feature was added to Cisco NX-OS to address table synchronization across vPC peers using the reliable transport mechanism of the Cisco Fabric Service over Ethernet (CFSoE) protocol. You must enable the `ip arp synchronize` and `ipv6 nd synchronize` commands to support faster convergence of address tables between the vPC peers. This convergence overcomes the delay that occurs in ARP table restoration for IPv4 or ND table restoration for IPv6 when the peer link port channel flaps or when a vPC peer comes back online.

vPC Multicast—PIM, IGMP, and IGMP Snooping

The Cisco NX-OS software for the Nexus 9000 Series devices supports the following on a vPC:

- PIM Any Source Multicast (ASM).
- PIM Source-Specific Multicast (SSM).

**Note**

The Cisco NX-OS software does not support Bidirectional (BIDR) on a vPC.

The software keeps the multicast forwarding state synchronized on both of the vPC peer devices. The IGMP snooping process on a vPC peer device shares the learned group information with the other vPC peer device through the vPC peer link; the multicast states are always synchronized on both vPC peer devices. The PIM process in vPC mode ensures that only one of the vPC peer devices forwards the multicast traffic to the receivers.

Each vPC peer is a Layer 2 or Layer 3 device. Multicast traffic flows from only one of the vPC peer devices. You might see duplicate packets in the following scenarios:

- Orphan hosts
• When the source and receivers are in the Layer 2 vPC cloud in different VLANs with multicast routing enabled and a vPC member link goes down.

You might see negligible traffic loss in the following scenarios:

• When you reload the vPC peer device that is forwarding the traffic.
• When you restart PIM on the vPC peer device that is forwarding the traffic.

Ensure that you dual-attach all Layer 3 devices to both vPC peer devices. If one vPC peer device goes down, the other vPC peer device continues to forward all multicast traffic normally.

The following outlines vPC PIM and vPC IGMP/IGMP snooping:

• vPC PIM—The PIM process in vPC mode ensures that only one vPC peer device forwards multicast traffic. The PIM process in vPC mode synchronizes the source state with both vPC peer devices and elects which vPC peer device forwards the traffic.

• vPC IGMP/IGMP snooping—The IGMP process in vPC mode synchronizes the designated router (DR) information on both vPC peer devices. Dual DRs are available for IGMP when you are in vPC mode. Dual DRs are not available when you are not in vPC mode, because both vPC peer devices maintain the multicast group information between the peers.

---

A PIM neighbor relationship between a vPC VLAN (a VLAN that is carried on a vPC peer link) and a downstream vPC-attached Layer 3 device is not supported, which can result in dropped multicast packets. If a PIM neighbor relationship is required with a downstream Layer 3 device, a physical Layer 3 interface must be used instead of a vPC interface.

You should enable or disable IGMP snooping identically on both vPC peer devices, and all the feature configurations should be identical. IGMP snooping is on by default.

---

The following commands are not supported in vPC mode:

• `ip pim spt-threshold infinity`

• `ip pim use-shared-tree-only`

See the Cisco Nexus 9000 Series NX-OS Multicast Routing Configuration Guide for more information about multicasting.

### Multicast PIM Dual DR (Proxy DR)

By default, a multicast router sends PIM joins upstream only if it has interested receivers. These interested receivers can either be IGMP hosts (they communicate through IGMP reports) or other multicast routers (they communicate through PIM joins).

In the Cisco NX-OS vPC implementation, PIM works in dual designated router (DR) mode. That is, if a vPC device is a DR on a vPC SVI outgoing interface (OIF), its peer automatically assumes the proxy DR role. IGMP adds an OIF (the report is learned on that OIF) to the forwarding if the OIF is a DR. With dual DRs, both vPC devices have an identical (*.G) entry with respect to the vPC SVI OIFs as shown in this example:
IP PIM PRE-BUILD SPT

When the multicast source is in a Layer 3 cloud (outside the vPC domain), one vPC peer is elected as the forwarder for the source. This forwarder election is based on the metrics to reach the source. If there is a tie, the vPC primary is chosen as the forwarder. Only the forwarder has the vPC OIFs in its associated (S,G) and the nonforwarder (S,G) has 0 OIFs. Therefore, only the forwarder sends PIM (S,G) joins toward the source as shown in this example:

VPC Device1 (say this is Forwarder for Source 'S'):
----------
(*,G)
oif1 (igmp)
(S,G)
oif1 (mrib)
VPC Device2:
----------
(*,G)
oif1 (igmp)
(S,G)
NULL

In the case of a failure (for example, a Layer 3 Reverse Path Forwarding (RPF) link on the forwarder becomes inoperable or the forwarder gets reloaded), if the current nonforwarder ends up becoming the forwarder, it has to start sending PIM joins for (S,G) toward the source to pull the traffic. Depending upon the number of hops to reach the source, this operation might take some time (PIM is a hop-by-hop protocol).

To eliminate this issue and get better convergence, use the ip pim pre-build-spt command. This command enables PIM send joins even if the multicast route has 0 OIFs. In a vPC device, the nonforwarder sends PIM (S,G) joins upstream toward the source. The downside is that the link bandwidth upstream from the nonforwarder gets used for the traffic that is ultimately dropped by it. The benefits that result with better convergence far outweigh the link bandwidth usage. Therefore, we recommend that you use this command if you use vPCs.

vPC Peer Links and Routing

The First Hop Redundancy Protocols (FHRPs) interoperate with vPCs. The Hot Standby Routing Protocol (HSRP), and Virtual Router Redundancy Protocol (VRRP) all interoperate with vPCs. We recommend that you dual-attach all Layer 3 devices to both vPC peer devices.

The primary FHRP device responds to ARP requests, even though the secondary vPC device forwards the data traffic.

To simplify initial configuration verification and vPC/HSRP troubleshooting, you can configure the primary vPC peer device with the FHRP active router highest priority.
In addition, you can use the priority command in the if-hsrp configuration mode to configure failover thresholds for when a group state enabled on a vPC peer link is in standby or in listen state. You can configure lower and upper thresholds to prevent the interface from going up and down.

VRRP acts similarly to HSRP when running on vPC peer devices. You should configure VRRP the same way that you configure HSRP.

When the primary vPC peer device fails over to the secondary vPC peer device, the FHRP traffic continues to flow seamlessly.

We recommend that you configure routing adjacency between the two vPC peer devices to act as a backup routing path. If one vPC peer device loses Layer 3 uplinks, the vPC can redirect the routed traffic to the other vPC peer device and leverage its active Layer 3 uplinks.

You can configure the inter-switch link for a backup routing path in the following ways:

- Create a Layer 3 link between the two vPC peer devices.
- Use the non-vPC VLAN trunk with a dedicated VLAN interface.
- Use a vPC peer link with a dedicated VLAN interface.

We do not recommend that you configure the burnt-in MAC address option (use-bia) for HSRP or manually configure virtual MAC addresses for any FHRP protocol in a vPC environment because these configurations can adversely affect vPC load balancing. The HSRP use-bia option is not supported on vPCs. When you are configuring custom MAC addresses, you must configure the same MAC address on both vPC peer devices.

You can use the delay restore command to configure a restore timer that delays the vPC coming back up until after the peer adjacency forms and the VLAN interfaces are back up. This feature enables you to avoid packet drops when the routing tables might not be converged before the vPC is once again passing traffic. Use the delay restore command to configure this feature.

To delay the VLAN interfaces on the restored vPC peer device from coming up, use the interfaces-vlan option of the delay restore command.

See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information about FHRPs and routing.

**Best Practices for Layer 3 and vPC Configuration**

This section describes best practices for using and configuring Layer 3 with vPC.

**Layer 3 and vPC Configuration Overview**

When a Layer 3 device is connected to a vPC domain through a vPC, it has the following views:

- At Layer 2, the Layer 3 device sees a unique Layer 2 switch presented by the vPC peer devices.
- At Layer 3, the Layer 3 device sees two distinct Layer 3 devices (one for each vPC peer device).

vPC is a Layer 2 virtualization technology, so at Layer 2, both vPC peer devices present themselves as a unique logical device to the rest of the network.

There is no virtualization technology at Layer 3, so each vPC peer device is seen as a distinct Layer 3 device by the rest of the network.

The following figure illustrates the two different Layer 2 and Layer 3 views with vPC.
Figure 17: Different Views for vPC Peer Devices

**Guidelines for Layer 3 and vPC Configurations**

To connect Layer 3 devices to a vPC domain, use Layer 3 links from Layer 3 devices to connect each vPC peer device.

*Note*

The vPC loop avoidance rule does not allow the attachment of a Layer 3 device to a vPC domain using a vPC.

Layer 3 devices are able to initiate Layer 3 routing protocol adjacencies with both vPC peer devices.

One or multiple Layer 3 links can be used to connect a Layer 3 device to each vPC peer device. Cisco Nexus 9000 series devices support Layer 3 Equal Cost Multipathing (ECMP) with up to 16 hardware load-sharing paths per prefix. Traffic from a vPC peer device to a Layer 3 device can be load-balanced across all the Layer 3 links interconnecting the two devices together.

Using Layer 3 ECMP on the Layer 3 device can effectively use all Layer 3 links from the device to the vPC domain. Traffic from a Layer 3 device to the vPC domain can be load-balanced across all the Layer 3 links interconnecting the two entities together.

The supported connection model for a Layer 3 device to the vPC domain is illustrated in the following figure.

**Figure 18: Using Separate Layer 3 Links to Connect L3 Device to a vPC Domain**

Follow these guidelines when connecting a Layer 3 device to the vPC domain:

- Use separate Layer 3 links to connect Layer 3 devices to the vPC domain.
- Do not use a Layer 2 vPC to attach a Layer 3 device to a vPC domain unless the Layer 3 device can statically route to the HSRP address configured on the vPC peer devices.

- When both routed and bridged traffic are required, use individual Layer 3 links for routed traffic and a separate Layer 2 port-channel for bridged traffic when both routed and bridged traffic are required.

- Enable Layer 3 connectivity between vPC peer devices by configuring a VLAN network interface for the same VLAN from both devices or by using a dedicated Layer 3 link between the two peer devices (for Layer 3 backup routing path purposes).

Example Topologies for Layer 3 and vPC

This section contains examples of network topologies for Layer 3 and vPC.

Figure 19: Legend

![Legend Diagram](image)

Peering Between Routers

In this example, vPC is used as a Layer 2 transit path. Because there is no direct routing protocol peering adjacency from the Layer 3 device to any vPC peer device, this topology is supported.

Figure 20: Peering Between Routers

![Topology Diagram](image)
**Peering with an External Router Using Layer 3 Links**

This example shows a topology that uses Layer 3 links to connect a Layer 3 device to the vPC domain.

---

**Note**

Interconnecting the 2 entities together in this way is a best practice.

---

**Figure 21: Peering with an External Router Using Layer 3 Links**

---

**Peering Between vPC Devices for a Backup Routing Path**

This example shows peering between the two vPC peer devices with a Layer 3 backup routed path. If the Layer 3 uplinks on vPC peer device 1 or vPC peer device 2 fail, the path between the two peer devices is used to redirect traffic to the switch that has the Layer 3 uplinks in the up state.

The Layer 3 backup routing path can be implemented using a dedicated interface VLAN (such as SVI) over the vPC peer-link or by using dedicated Layer 2 or Layer 3 links across the two vPC peer devices.

---

**Figure 22: Peering Between vPC Devices for a Backup Routing Path**

---

**Peering Between Two Routers with vPC Devices as Transit Switches**

This example is similar to the Peering between Routers topology. The difference here is that the vPC domains are only used as Layer 2 transit paths.
Peering with an External Router on Parallel Interconnected Routed Ports

This example shows the Layer 3 device attached to the vPC domain through two different types of links, Layer 2 links and Layer 3 links.

The Layer 2 links are used for bridged traffic (traffic staying in the same VLAN) or inter-VLAN traffic (assuming vPC domain hosts the interface VLAN and associated HSRP configuration).

The Layer 3 links are used for routing protocol peering adjacency with each vPC peer device.

The purpose of this topology is to attract specific traffic to go through the Layer 3 device. Layer 3 links are also used to carry routed traffic from a Layer 3 device to the vPC domain.

Peering Over a vPC Interconnection on Parallel Interconnected Routed Ports

When routing protocol peering adjacency is required to be established between the two data centers, a best practice is to add dedicated Layer 3 links between the two sites as shown in this example.

The vPC link between the two data centers carry bridged traffic or inter-VLAN traffic while the dedicated Layer 3 links carry the routed traffic across the two sites.
This example shows when the Layer 3 device is single-attached to the vPC domain, you can use a non-vPC VLAN with a dedicated inter-switch link to establish the routing protocol peering adjacency between the Layer 3 device and each vPC peer device. However, the non-vPC VLAN must be configured to use a static MAC that is different than the vPC VLAN.

**Note**
Configuring the vPC VLAN (and vPC peer-link) for this purpose is not supported.

**CFSoE**

The Cisco Fabric Services over Ethernet (CFSoE) is a reliable state transport mechanism that is used to synchronize the actions of the vPC peer devices. CFSoE carries messages and packets for many features linked with vPC, such as STP and IGMP. Information is carried in CFS/CFSoE protocol data units (PDUs).

When you enable the vPC feature, the device automatically enables CFSoE, and you do not have to configure anything. CFSoE distributions for vPCs do not need the capabilities to distribute over IP or the CFS regions. You do not need to configure anything for the CFSoE feature to work correctly on vPCs.

The CFSoE transport is local to each VDC.
You can use the `show mac address-table` command to display the MAC addresses that CFSoE synchronizes for the vPC peer link.

**Note**

Do not enter the `no cfs eth distribute` or the `no cfs distribute` command. You must enable CFSoE for vPC functionality. If you do enter either of these commands with vPC enabled, the system displays an error message.

When you enter the `show cfs application` command, the output displays “Physical-eth,” which shows the applications that are using CFSoE.

CFS also transports data over TCP/IP. See the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide for more information about CFS over IP.

**Note**

The software does not support CFS regions.

### vPC and Orphan Ports

When a device that is not vPC-capable connects to each peer, the connected ports are known as orphan ports because they are not members of a vPC. The device’s link to one peer will be active (forwarding) and the other link will be standby (blocking) due to STP.

If a peer link failure or restoration occurs, an orphan port’s connectivity might be bound to the vPC failure or restoration process. For example, if a device’s active orphan port connects to the secondary vPC peer, the device loses any connections through the primary peer if a peer link failure occurs and the vPC ports are suspended by the secondary peer. If the secondary peer were to also suspend the active orphan port, the device’s standby port becomes active, provides a connection to the primary peer, and restores connectivity.

You can configure in the CLI that specific orphan ports are suspended by the secondary peer when it suspends its vPC ports and are restored when the vPC is restored.

### Virtualization Support

All ports in a given vPC must be in the same VDC. This version of the software supports only one vPC domain per VDC. You can use the numbers from 1 to 4096 in each VDC to number the vPC.

### vPC Recovery After an Outage

In a data center outage, both of the Cisco Nexus 9000 Series devices that include a vPC get reloaded. Occasionally only one peer can be restored. With no functioning peer-keepalive or peer link, the vPC cannot function normally, but depending on your Cisco NX-OS release, a method might be available to allow vPC services to use only the local ports of the functional peer.

### Autorecovery

You can configure the Cisco Nexus 9000 Series device to restore vPC services when its peer fails to come online by using the `auto-recovery` command. You must save this setting in the startup configuration. On reload, if the peer link is down and three consecutive peer-keepalive messages are lost, the secondary device assumes the primary STP role and the primary LACP role. The software reinitializes the vPCs, bringing up its local ports. Because there are no peers, the consistency check is bypassed for the local vPC ports. The
device elects itself to be the STP primary regardless of its role priority and also acts as the master for LACP port roles.

vPC Peer Roles After a Recovery

When the other peer device completes its reload and adjacency forms, the following process occurs:

1. The first vPC peer maintains its current role to avoid any transition reset to other protocols. The peer accepts the other available role.
2. When an adjacency forms, consistency checks are performed and appropriate actions are taken.

High Availability

During an In-Service Software Upgrade (ISSU), the software reload process on the first vPC device locks its vPC peer device by using CFS messaging over the vPC communications channel. Only one device at a time is upgraded. When the first device completes its upgrade, it unlocks its peer device. The second device then performs the upgrade process, locking the first device as it does so. During the upgrade, the two vPC devices temporarily run different releases of Cisco NX-OS, however the system functions correctly because of its backward compatibility support.

Note
See the Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide for complete information about high-availability features.

vPC Forklift Upgrade Scenario

The following describes a scenario for migrating from a pair of Cisco Nexus 9000 Series switches in a vPC topology to a different pair of Cisco Nexus 9000 Series switches. Typically the scenario might be migrating from a pair of Cisco Nexus 9508 vPC peer nodes to a pair of Cisco Nexus 9516 switches.

Considerations for a vPC forklift upgrade:

• vPC Role Election and Sticky-bit
  When the two vPC systems are joined to form a vPC domain, priority decides which device is the vPC primary and which is the vPC secondary. When the primary device is reloaded, the system comes back online and connectivity to the vPC secondary device (now the operational primary) is restored. The operational role of the secondary device (operational primary) does not change (to avoid unnecessary disruptions). This behavior is achieved with a sticky-bit, where the sticky information is not saved in the startup configuration. This method makes the device that is up and running win over the reloaded device. Hence, the vPC primary becomes the vPC operational secondary. Sticky-bit is also set when a vPC node comes up with peer-link and peer-keepalive down and it becomes primary after the auto recovery period.

• vPC Delay Restore
  The delay restore timer is used to delay the vPC from coming up on the restored vPC peer device after a reload when the peer adjacency is already established.
  To delay the VLAN interfaces on the restored vPC peer device from coming up, use the `interfaces-vlan` option of the `delay restore` command.
vPC Auto-Recovery

During a data center power outage when both vPC peer switches go down, if only one switch is restored, the auto-recovery feature allows that switch to assume the role of the primary switch and the vPC links come up after the auto-recovery time period. The default auto-recovery period is 240 seconds.

The following example is a migration scenario that replaces vPC peer nodes Node1 and Node2 with New_Node1 and New_Node2.

<table>
<thead>
<tr>
<th>Migration Step</th>
<th>Expected Behavior</th>
<th>Node1 Configured role (Ex: role priority 100)</th>
<th>Node1 Operational role</th>
<th>Node2 Configured role (Ex: role priority 200)</th>
<th>Node2 Operational role</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initial state</td>
<td>Traffic is forwarded by both vPC peers – Node1 and Node2. Node1 is primary and Node2 is secondary.</td>
<td>primary</td>
<td>Primary Sticky bit: False</td>
<td>secondary</td>
</tr>
<tr>
<td>2</td>
<td>Node2 replacement – Shut all vPCs and uplinks on Node2. Peer-link and vPC peer-keepalive are in administrative up state.</td>
<td>Traffic converged on Primary vPC peer Node1.</td>
<td>primary</td>
<td>Primary Sticky bit: False</td>
<td>secondary</td>
</tr>
<tr>
<td>3</td>
<td>Remove Node2.</td>
<td>Node1 will continue to forward traffic.</td>
<td>primary</td>
<td>Primary Sticky bit: False</td>
<td>n/a</td>
</tr>
<tr>
<td>4</td>
<td>Configure New_Node2. Copy the configuration to startup config. vPC peer-link and peer-keepalive in administrative up state. Power off New_Node2. Make all connections. Power on New_Node2.</td>
<td>New_Node2 will come up as secondary. Node1 continue to be primary. Traffic will continue to be forwarded on Node01.</td>
<td>primary</td>
<td>Primary Sticky bit: False</td>
<td>secondary</td>
</tr>
<tr>
<td>Migration Step</td>
<td>Expected Behavior</td>
<td>Node1 Configured role (Ex: role priority 100)</td>
<td>Node1 Operational role</td>
<td>Node2 Configured role (Ex: role priority 200)</td>
<td>Node2 Operational role</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------</td>
<td>---------------------------------------------</td>
<td>------------------------</td>
<td>---------------------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>5</td>
<td>Bring up all vPCs and uplink ports on New_Node2.</td>
<td>Traffic will be forwarded by both Node 1 and New_Node2.</td>
<td>primary</td>
<td>Primary Sticky bit: False</td>
<td>secondary</td>
</tr>
<tr>
<td>6</td>
<td>Node1 replacement - Shut vPCs and uplinks on Node1.</td>
<td>Traffic will converge on New_Node2.</td>
<td>primary</td>
<td>Primary Sticky bit: False</td>
<td>secondary</td>
</tr>
<tr>
<td>7</td>
<td>Remove Node1.</td>
<td>New_Node2 will become secondary, operational primary and sticky bit will be set to True.</td>
<td>n/a</td>
<td>n/a</td>
<td>secondary</td>
</tr>
<tr>
<td>9</td>
<td>Bring up all vPCs and uplink ports on New_Node1.</td>
<td>Traffic will be forwarded by both New_Node1 and New_Node2.</td>
<td>primary</td>
<td>Secondary Sticky bit: False</td>
<td>secondary</td>
</tr>
</tbody>
</table>

**Note**

If you prefer to have the configured secondary node as the operational secondary and the configured primary as the operational primary, then Node2 can be reloaded at the end of the migration. This is optional and does not have any functional impact.

**Licensing Requirements for vPCs**

The following table shows the licensing requirements for this feature:
Product | License Requirement
--- | ---
Cisco NX-OS | Cisco NX-OS vPCs require no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you.

**Guidelines and Limitations**

vPCs have the following configuration guidelines and limitations:

- Make sure that both vPC peers are in the same mode (regular mode or enhanced mode) before performing a nondisruptive upgrade.

**Note**

vPC peering between an enhanced ISSU mode (boot mode lxc) configured switch and a non-enhanced ISSU mode switch is not supported.

- `show` commands with the `internal` keyword are not supported.
- Cisco Nexus 9000 Series switches do not support NAT on vPC topology.
- The `spanning-tree pseudo-information` command is not available on Cisco Nexus 92160 and Cisco Nexus 93180 switches.
- The `show vpc consistency-checker` command is not available on Cisco Nexus 9000 switches starting from Cisco NX-OS Release 9.2(1).
- The `delay restore interface-bridge-domain` and `peer-gateway exclude-bridge-domain` commands are not available on Cisco Nexus 9500-R platform switches starting from Cisco NX-OS Release 9.2(1).
- vPC peers must run the same Cisco NX-OS release. During a software upgrade, make sure to upgrade the primary vPC peer first.
- All ports for a given vPC must be in the same VDC.
- You must enable vPCs before you can configure them.
- You must configure the peer-keapalive link and messages before the system can form the vPC peer link.
- Only Layer 2 port channels can be in vPCs.
- You must configure both vPC peer devices; the configuration is not sent from one device to the other.
- To configure multilayer (back-to-back) vPCs, you must assign unique vPC domain ID for each respective vPC.
- Check that the necessary configuration parameters are compatible on both sides of the vPC peer link. See the “Compatibility Parameters for vPC Interfaces” section for information about compatibility recommendations.
- You may experience minimal traffic disruption while configuring vPCs.
- The software does not support BIDR PIM on vPCs.
• The software does not support CFS regions.

• Port security is not supported on port channels.

• When **peer-switch** features are configured under **vpc domain** configuration mode on two Cisco Nexus 9000 Series switches, the spanning-tree root changes even for VLANs that are not enabled on the vPC peer-link. Both the switches act as one system with one MAC address as the bridge address. This is true even for non-vPC mst-instance or VLANs. Therefore, a non vPC peer-link between the two switches gets blocked as a backup link. This is an expected behavior.

• We recommend that you configure all the port channels in the vPC using LACP with the interfaces in active mode.

• Back-to-back, multilayer vPC topologies require unique domain IDs on each respective vPC.

• Having the same Hot Standby Router Protocol (HSRP)/Virtual Router Redundancy Protocol (VRRP) group on all nodes on a double sided vPC is supported.

• When migrating from a pair of spine nodes to a pair of Cisco Nexus 9000 devices, the HSRP priority should be configured so that the Cisco Nexus 9000 vPC peers are in Active/Standby state. There is no support for Cisco Nexus 9000 vPC peers in HSRP state to be in Active/Listen state, or Standby/Listen state.

• When using vPCs, we recommend that you use default timers for FHRP (HSRP, VRRP), and PIM configurations. There is no advantage in convergence times when using aggressive timers in vPC configurations.

• If you configure open shortest path first (OSPF) in a vPC environment, use the following timer commands in router configuration mode on the core switch to ensure fast OSPF convergence when a vPC peer link is shut down:

```
switch (config-router)# timers throttle spf 1 50 50
switch (config-router)# timers lsa-arrival 10
```

See the *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide* for further details about OSPF.

• BFD for VRRP/HSRP is not supported in a vPC environment.

• The STP port cost is fixed to 200 in a vPC environment.

• Jumbo frames are enabled by default on the vPC peer link.

• To accommodate increased traffic when the vPC goes down and traffic needs to cross the peer-link, it is a best practice to use multiple high bandwidth interfaces (such as the 40G interfaces for the Cisco Nexus 9000) across linecards for the peer-link.

• The **vpc orphan-ports suspend** command also applies to ports in non-vPC VLANs and Layer 3 ports. However, it is recommended to be used with ports in VPC VLANs.

• FEX-AA (dual-homed FEX) and FEX-ST (FEX straight-thru) topologies (FEX-AA and FEX-ST) are supported. The following mixing is not supported as the parent switches:
  
  • Cisco Nexus 9300-EX and 9300 switches
  
  • Cisco Nexus 9300 and 9500 switches
  
  • Cisco Nexus 9300-EX and 9500 switches
• When configuring vPCs, the behavior previously provided by using the ip pim pre-build-spt command has now been enabled automatically by default and cannot be disabled.

• A vPC port channel member link that is operating in Individual state will be flapped while checking for VLAN inconsistencies. To avoid having the link flapped during server provisioning, disable the VPC graceful consistency check with the no graceful consistency-check command.

The following example disables the VPC graceful consistency check:

switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# vpc domain 1
switch(config-vpc-domain)# no graceful consistency-check

• vPC STP hitless role change feature is supported.
• vPC role change can be performed from either of the peer devices.
• If the original secondary device has higher role priority value than the original primary device, role swapping cannot be performed. Change the role priority on either vPC device so that the value of the original secondary device is lower than the original primary one. To view the existing role of a device, use the show vpc role command on local and peer switch.
• You can connect a pair of switches such as Cisco Nexus 9300 Series switches or Cisco Nexus 9500 Series switches in a vPC domain, that will for a vPC to a network node or to a host.
  • Switches in a vPC domain, peer switches must be of the same type, for example, you can connect a pair of Cisco Nexus 9300 series switches, but you cannot connect a Cisco Nexus 9300 series switch to a Cisco Nexus 9500 series switch in a vPC domain.
  • vPC peer switch families must also be of the same type, for example, you can connect a pair of Cisco Nexus 9300-EX switches, but you cannot connect a Cisco Nexus 9300-EX switch to a Cisco Nexus 9300-FX switch in a vPC domain.
• Always check the existing configured role priority before configuring vPC hitless role change feature
• In a vPC domain, enable the peer-switch command, where both vPC peers have same STP priorities, and ensure it is operational before issuing a role change. If you do not enable the peer-switch command, it can lead to convergence issues. Use show spanning-tree summary | grep peer command to verify whether the peer vPC switch is operational or not.
• All the devices that are attached to a vPC domain must be dual homed.

Default Settings

The following table lists the default settings for vPC parameters.

Table 13: Default vPC Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>vPC system priority</td>
<td>32667</td>
</tr>
</tbody>
</table>
### Configuring vPCs

#### Note
You must use these procedures on both devices on both sides of the vPC peer link. You configure both of the vPC peer devices using these procedures.

This section describes how to configure vPCs using the command-line interface (CLI).

#### Note
If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

### Enabling vPCs

You must enable the vPC functionality before you can configure and use vPCs.

#### SUMMARY STEPS

1. `configure terminal`
2. `feature vpc`
3. `exit`
4. `show feature`
5. `copy running-config startup-config`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config)#</code></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>feature vpc</code></td>
<td>Enables vPCs on the device.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config)# feature vpc</code></td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

#### Command or Action

<table>
<thead>
<tr>
<th>Step 3</th>
<th>exit</th>
</tr>
</thead>
</table>

**Example:**

```
switch(config)# exit
switch#
```

<table>
<thead>
<tr>
<th>Step 4</th>
<th>show feature</th>
</tr>
</thead>
</table>

**Example:**

```
switch# show feature
```

<table>
<thead>
<tr>
<th>Step 5</th>
<th>copy running-config startup-config</th>
</tr>
</thead>
</table>

**Example:**

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

### Example

This example shows how to enable the vPC feature:

```
switch# configure terminal
switch(config)# feature vpc
switch(config)# exit
switch(config)#
```

### Disabling vPCs

When you disable the vPC functionality, the device clears all the vPC configurations.

### SUMMARY STEPS

1. configure terminal
2. no feature vpc
3. exit
4. show feature
5. copy running-config startup-config

### Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure terminal</td>
</tr>
</tbody>
</table>

**Example:**

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

<table>
<thead>
<tr>
<th>Step 3</th>
<th>exit</th>
</tr>
</thead>
</table>

**Example:**

```
switch(config)# exit
switch(config)#
```
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>no feature vpc</strong>  &lt;br&gt;<strong>Example:</strong>  &lt;br&gt;switch(config)# no feature vpc</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>exit</strong>  &lt;br&gt;<strong>Example:</strong>  &lt;br&gt;switch(config)# exit  &lt;br&gt;switch#</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>show feature</strong>  &lt;br&gt;<strong>Example:</strong>  &lt;br&gt;switch# show feature</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>copy running-config startup-config</strong>  &lt;br&gt;<strong>Example:</strong>  &lt;br&gt;switch# copy running-config startup-config</td>
</tr>
</tbody>
</table>

**Example**

This example shows how to disable the vPC feature:

```
switch# configure terminal
switch(config)# no feature vpc
switch(config)# exit
switch#
```

**Creating a vPC Domain and Entering vpc-domain Mode**

You can create a vPC domain and put the vPC peer link port channels into the identical vPC domain on both vPC peer devices. Use a unique vPC domain number throughout a single vPC domain. This domain ID is used to automatically to form the vPC system MAC address.

You can also use this command to enter vpc-domain command mode.

**SUMMARY STEPS**

1. `configure terminal`
2. `vpc domain domain-id [shut | no shut]`
3. `exit`
4. `show vpc brief`
5. `copy running-config startup-config`
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config)#</code></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>`vpc domain domain-id [shut</td>
<td>no shut]`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config)# vpc domain 5</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-vpc-domain)#</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>exit</code></td>
<td>Exits vpc-domain configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config)# exit</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch#</code></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>show vpc brief</code></td>
<td>(Optional) Displays brief information about each vPC domain.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch# show vpc brief</code></td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td><code>copy running-config startup-config</code></td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch# copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>

### Example

This example shows how to enter the vpc-domain command mode to configure an existing vPC domain:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# exit
switch(config)#
```

## Configuring a vPC Keepalive Link and Messages

### Note

You must configure the vPC peer-keepalive link before the system can form the vPC peer link.

You can configure the destination IP for the peer-keepalive link that carries the keepalive messages. Optionally, you can configure other parameters for the keepalive messages.
We recommend that you configure a separate VRF instance and put a Layer 3 port from each vPC peer device into that VRF for the vPC peer-keepalive link. Do not use the peer link itself to send vPC peer-keepalive messages. For information about creating and configuring VRFs, see the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide. Ensure that both the source and destination IP addresses use for the peer-keepalive message are unique in your network. The management port and management VRF are the defaults for these keepalive messages.

**Before you begin**

Ensure that you have enabled the vPC feature.

**SUMMARY STEPS**

1. `configure terminal`
2. `vpc domain domain-id [shut | no shut]`
3. `peer-keepalive destination ipaddress [hold-timeout secs | interval msecs {timeout secs} | {precedence {prec-value | network | internet | critical | flash-override | flash | immediate priority | routine} | tos {tos-value | max-reliability | max-throughput | min-delay | min-monetary-cost | normal} | tos-byte {tos-byte-value | source ipaddress | vrf {name | management vpc-keepalive}]]`
4. `exit`
5. `show vpc statistics`
6. `copy running-config startup-config`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> switch# configure terminal switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> vpc domain domain-id [shut</td>
<td>no shut]</td>
</tr>
<tr>
<td><strong>Example:</strong> switch(config)# vpc domain 5 switch(config-vpc-domain)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> peer-keepalive destination ipaddress [hold-timeout secs</td>
<td>interval msecs {timeout secs}</td>
</tr>
<tr>
<td><strong>Note</strong> The system does not form the vPC peer link until you configure a vPC peer-keepalive link.</td>
<td></td>
</tr>
<tr>
<td>The management ports and VRF are the defaults.</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring vPCs

#### Creating a vPC Peer Link

**You create the vPC peer link by designating the port channel that you want on each device as the peer link for the specified vPC domain. We recommend that you configure the Layer 2 port channels that you are designating as the vPC peer link in trunk mode and that you use two ports on separate modules on each vPC peer device for redundancy.**

**Before you begin**

Ensure that you have enabled the vPC feature.

**Example**

For more information about configuring VRFs, see the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.

This example shows how to configure the destination and source IP address and VRF for the vPC-peer-keepalive link:

```plaintext
switch# configure terminal
switch(config)# vpc domain 100
switch(config-vpc-domain)# peer-keepalive destination 172.168.1.2 source 172.168.1.1 vrf vpc-keepalive
switch(config-vpc-domain)# exit
switch#
```

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td>switch(config-vpc-domain)# peer-keepalive destination 172.28.230.85 switch(config-vpc-domain)#</td>
<td>We recommend that you configure a separate VRF and use a Layer 3 port from each vPC peer device in that VRF for the vPC peer-keepalive link. For more information about creating and configuring VRFs, see the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.</td>
</tr>
<tr>
<td>exit</td>
<td>Exits global configuration mode.</td>
</tr>
<tr>
<td>show vpc statistics</td>
<td>(Optional) Displays information about the configuration for the keepalive messages.</td>
</tr>
<tr>
<td>copy running-config startup-config</td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>

---

**Step 4**

**exit**

**Example:**

```plaintext
switch(config)# exit
switch#
```

**Step 5**

**show vpc statistics**

**Example:**

```plaintext
switch# show vpc statistics
```

**Step 6**

**copy running-config startup-config**

**Example:**

```plaintext
switch# copy running-config startup-config
```
### SUMMARY STEPS

1. configure terminal
2. interface port-channel *channel-number*
3. switchport mode trunk
4. switchport trunk allowed vlan *vlan-list*
5. vpc peer-link
6. exit
7. show vpc brief
8. copy running-config startup-config

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** configure terminal  
Example:  
```
switch# configure terminal
switch(config)#
``` | Enters global configuration mode. |
| **Step 2** interface port-channel *channel-number*  
Example:  
```
switch(config)# interface port-channel 20
switch(config-if)#
``` | Selects the port channel that you want to use as the vPC peer link for this device, and enters interface configuration mode. |
| **Step 3** switchport mode trunk  
Example:  
```
switch(config-if)# switchport mode trunk
``` | (Optional) Configures this interface in trunk mode. |
| **Step 4** switchport trunk allowed vlan *vlan-list*  
Example:  
```
switch(config-if)# switchport trunk allowed vlan 1-120,201-3967
``` | (Optional) Configures the permitted VLAN list. |
| **Step 5** vpc peer-link  
Example:  
```
switch(config-if)# vpc peer-link
switch(config-vpc-domain)#
``` | Configures the selected port channel as the vPC peer link, and enters vpc-domain configuration mode. |
| **Step 6** exit  
Example:  
```
switch(config)# exit
switch#
``` | Exits vpc-domain configuration mode. |
| **Step 7** show vpc brief  
Example:  
```
switch# show vpc brief
``` | (Optional) Displays information about each vPC, including information about the vPC peer link. |
Example

This example shows how to configure a vPC peer link:

```
switch# configure terminal
switch(config)# interface port-channel 20
switch(config-if)# switchport mode
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk allowed vlan 1-120,201-3967
switch(config-if)# vpc peer-link
switch(config-vpc-domain)# exit
switch(config)#
```

Configuring a vPC Peer-Gateway

You can configure vPC peer devices to act as the gateway for packets that are destined to the vPC peer device's MAC address.

Before you begin

Ensure that you have enabled the vPC feature.

SUMMARY STEPS

1. configure terminal
2. vpc domain domain-id [shut | no shut]
3. peer-gateway
4. exit
5. show vpc brief
6. copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 2**                         |                                              |
| vpc domain domain-id [shut | no shut] | Creates a vPC domain if it does not already exist, and enters vpc-domain configuration mode. |
| Example:                           |                                              |
| switch(config-if)# vpc domain 5    |                                              |
| switch(config-vpc-domain)#        |                                              |
Purpose
Command or Action

Step 3 peer-gateway
Example:
switch(config-vpc-domain)# peer-gateway
Note Disable IP redirects on all interface-vlans of this vPC domain for correct operation of this feature.

Step 4 exit
Example:
switch(config)# exit
switch#

Step 5 show vpc brief
Example:
switch# show vpc brief

Step 6 copy running-config startup-config
Example:
switch# copy running-config startup-config

Configuring Fast Convergence

Fast convergence feature is supported on Cisco Nexus 9000 Series platforms. You can enable or disable the vPC optimizations using this command. To achieve faster convergence, you must enable [no] fast-convergence on both vPC peers to achieve fast-convergence.

SUMMARY STEPS

1. configure terminal
2. switch(config) # vpc domain <domain>
3. switch(config) # peer-switch
4. switch(config) # show vpc peer-keepalive
5. switch(config) # delay restore { time }
6. switch(config) # peer-gateway
7. switch(config) # delay restore orphan-port
8. switch(config-vpc-domain)# fast-convergence

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2 switch(config) # vpc domain &lt;domain&gt;</td>
<td>Configure the VPC domain number.</td>
</tr>
<tr>
<td>Step 3 switch(config) # peer-switch</td>
<td>Define the peer switch.</td>
</tr>
</tbody>
</table>
### Configuring vPCs

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Command or Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Displays information about the peer keepalive messages</td>
<td><code>switch(config) # show vpc peer-keepalive</code></td>
</tr>
<tr>
<td>Number of seconds to delay bringing up the restored vPC peer device. The range is from 1 to 3600.</td>
<td><code>switch(config) # delay restore { time }</code></td>
</tr>
<tr>
<td>To enable Layer 3 forwarding for packets destined to the gateway MAC address of the virtual Port Channel (vPC), use the peer-gateway command. To disable Layer 3 forwarding packets, use the no form of this command.</td>
<td><code>switch(config) # peer-gateway</code></td>
</tr>
<tr>
<td>Number of seconds to delay bringing up the restored device's orphan port</td>
<td><code>switch(config) # delay restore orphan-port</code></td>
</tr>
<tr>
<td>Configure vPC fast convergence.</td>
<td><code>switch(config-vpc-domain)# fast-convergence</code></td>
</tr>
</tbody>
</table>

### Configuring LACP vPC Convergence

Beginning with Cisco NX-OS Release 7.0(3)I7(5), Link Aggregation Control Protocol (LACP) vPC convergence feature is supported on Cisco Nexus 9500 Series Switches with 9700-EX and 9700-FX line cards. This feature is not supported on Nexus 9500 with 9400, 9500, and 9600 and 9600-R line cards.

Link Aggregation Control Protocol (LACP) vPC convergence feature is supported on Cisco Nexus 9500 Series Switches with 9700-EX and 9700-FX line cards. This feature is not supported on Nexus 9500 with 9400, 9500, and 9600 and 9600-R line cards.

Link Aggregation Control Protocol (LACP) vPC convergence feature is supported on Cisco Nexus 9200 and 9300 Series Switches. You can configure LACP vPC convergence feature for more efficient use of port channels by reducing convergence time of vPC port channel for member link going down and first member bring up.

When you configure LACP vPC convergence on a Cisco Nexus 9000 switch, it waits until all the VLANs are initialized and programmed and then send LACP sync PDU, which will start sending traffic to the VPC domain without drops. You may configure the `lacp vpc-convergence` command in a VXLAN and non-VXLAN environments that have vPC port-channels to hosts that support LACP.

### SUMMARY STEPS

1. `configure terminal`
2. `switch(config) # interface {type/slot | portchannel number}`
3. `switch(config-if) # lacp vpc-convergence`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Command or Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enters global configuration mode.</td>
<td><code>configure terminal</code></td>
</tr>
<tr>
<td>Specifies an interface to configure, and enters interface configuration mode.</td>
<td>`switch(config) # interface {type/slot</td>
</tr>
</tbody>
</table>
### Configuring Layer 3 over vPC

Layer 3 over vPC is supported on Cisco Nexus 9000 Series switches.

**Before you begin**

Ensure that the peer-gateway feature is enabled and it is configured on both the peers and both the peers run an image that supports Layer 3 over vPC. If you enter the `layer3 peer-router` command without enabling the peer-gateway feature, a syslog message is displayed recommending you to enable the peer-gateway feature. Ensure that the peer link is up.

**SUMMARY STEPS**

1. switch# configure terminal
2. switch(config)# vpc domain domain-id
3. switch(config-vpc-domain)#layer3 peer-router
4. switch(config-vpc-domain)# exit
5. (Optional) switch#show vpc brief
6. (Optional) switch#copy running-config startup-config
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Creates a vPC domain if it does not already exist, and enters the vpc-domain configuration mode. There is no default; the range is from &lt;1 to 1000&gt;.</td>
</tr>
<tr>
<td>switch(config)# vpc domain domain-id</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# vpc domain 5</td>
<td></td>
</tr>
<tr>
<td>switch(config-vpc-domain)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enables the Layer 3 device to form peering adjacency with both the peers.</td>
</tr>
<tr>
<td>switch(config-vpc-domain)# layer3 peer-router</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Configure this command in both the peers. If you configure this command only on one of the peers or you disable it on one peer, the operational state of layer 3 peer-router gets disabled. You get a notification when there is a change in the operational state.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Exits the vpc-domain configuration mode.</td>
</tr>
<tr>
<td>switch(config-vpc-domain)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Displays brief information about each vPC domain.</td>
</tr>
<tr>
<td>(Optional) switch# show vpc brief</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td>(Optional) switch# copy running-config startup-config</td>
<td></td>
</tr>
</tbody>
</table>

### Example

The following example shows how to configure Layer 3 over vPC feature:

```bash
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# layer3 peer-router
switch(config-vpc-domain)# exit
switch(config)#
```

This example shows how to verify if the Layer 3 over vPC feature is configured. The **Operational Layer3 Peer** is enabled or disabled depending up on how the operational state of Layer 3 over vPC is configured.

```bash
switch# show vpc brief

vPC domain id : 5
Peer status : peer adjacency formed ok
vPC keep-alive status : peer is alive
Configuration consistency status : success
Per-vlan consistency status : failed
```
Type-2 consistency status: success
vPC role: secondary
Number of vPCs configured: 2
Peer Gateway: Enabled
Peer gateway excluded VLANs: -
Dual-active excluded VLANs: -
Graceful Consistency Check: Enabled
Auto-recovery status: Enabled (timeout = 240 seconds)
Operational Layer3 Peer: Enabled

Layer 3 over vPC Supported Topologies

See the following Layer 3 over vPC supported topologies:

*Figure 27: Supported: Peering Over a vPC Interconnection Where the Router Peers with Both the vPC Peers.*
Figure 28: Supported: Peering Over an STP Interconnection Using a vPC VLAN Where the Router Peers with Both the vPC Peers.

Figure 29: Supported: Peering Over an Orphan Device with Both the vPC Peers.
Configuring a Graceful Consistency Check

You can configure the graceful consistency check feature, which is enabled by default. Unless this feature is enabled, the vPC is completely suspended when a mismatch in a mandatory compatibility parameter is introduced in a working vPC. When this feature is enabled, only the links on the secondary peer device are suspended. See the “Compatibility Parameters for vPC Interfaces” section for information about consistent configurations on the vPCs.

SUMMARY STEPS

1. configure terminal
2. vpc domain domain-id [shut | no shut]
3. graceful consistency-check
4. exit
5. show vpc brief

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure terminal</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>Creates a vPC domain if it does not already exist, and enters vpc-domain configuration mode.</td>
</tr>
<tr>
<td>vpc domain domain-id [shut</td>
<td>no shut]</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring vPCs

#### Checking the Configuration Compatibility on a vPC Peer Link

After you have configured the vPC peer link on both vPC peer devices, check that the configurations are consistent on all vPC interfaces. See the “Compatibility Parameters for vPC Interfaces” section for information about consistent configurations on the vPCs.

**SUMMARY STEPS**

1. configure terminal
2. show vpc consistency-parameters {global | interface port-channel channel-number}

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> switch# configure terminal switch(config)#</td>
<td></td>
</tr>
</tbody>
</table>

---

### Example

This example shows how to enable the graceful consistency check feature:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# graceful consistency-check
switch(config-vpc-domain)# exit
switch(config)#
```

---

### Checking the Configuration Compatibility on a vPC Peer Link

#### Command or Action

```
switch(config-if)# vpc domain 5
switch(config-vpc-domain)#
```

#### Purpose

Specifies that only the links on the secondary peer device are suspended when a mismatch is detected in a mandatory compatibility parameter.

Use the `no` form of this command to disable the feature.

**Step 3**

**Graceful Consistency-check**

**Example:**

```
switch(config-vpc-domain)# graceful consistency-check
```

**Step 4**

**exit**

**Example:**

```
switch(config)# exit
switch#
```

**Step 5**

**show vpc brief**

**Example:**

```
switch# show vpc brief
```
### Moving Other Port Channels into a vPC

We recommend that you attach the vPC domain downstream port channel to two devices for redundancy.

To connect to the downstream device, you create a port channel from the downstream device to the primary vPC peer device and you create another port channel from the downstream device to the secondary peer device. On each vPC peer device, you assign a vPC number to the port channel that connects to the downstream device. You will experience minimal traffic disruption when you are creating vPCs.

Before you begin

Ensure that you have enabled the vPC feature.

Ensure that you are using a Layer 2 port channel.

**SUMMARY STEPS**

1. configure terminal
2. interface port-channel *channel-number*
3. vpc *number*
4. exit
5. show vpc brief
6. copy running-config startup-config

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong></td>
<td>(Optional) Displays the status of those parameters that must be consistent across all vPC interfaces.</td>
</tr>
<tr>
<td>`show vpc consistency-parameters {global</td>
<td>interface port-channel <em>channel-number</em>}`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>switch(config)# show vpc consistency-parameters global</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config)#</code></td>
<td></td>
</tr>
</tbody>
</table>

---

**Note**

Messages regarding the vPC interface configuration compatibility are also logged to the syslog.
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>interface port-channel channel-number</td>
<td>Selects the port channel that you want to put into the vPC to connect to the downstream device, and enters interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)# interface port-channel 20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>vpc number</td>
<td>Configures the selected port channel into the vPC to connect to the downstream device. You can use any module in the device for these port channels. The range is from 1 and 4096.</td>
</tr>
<tr>
<td></td>
<td>Note</td>
<td>The vPC number that you assign to the port channel connecting to the downstream device from the vPC peer device must be identical on both vPC peer devices.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# vpc 5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-vpc-domain)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>exit</td>
<td>Exits vpc-domain configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)# exit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>show vpc brief</td>
<td>(Optional) Displays information on the vPCs.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch# show vpc brief</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>copy running-config startup-config</td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch# copy running-config startup-config</td>
<td></td>
</tr>
</tbody>
</table>

### Example

This example shows how to configure a port channel to connect to the downstream device:

```
switch# configure terminal
switch(config)# interface port-channel 20
switch(config-if)# vpc 5
switch(config-if)# exit
switch(config)#
```
Manually Configuring a vPC Domain MAC Address

When you create a vPC domain, the Cisco NX-OS software automatically creates a vPC system MAC address, which is used for operations that are confined to the link-scope, such as LACP. However, you might choose to configure the vPC domain MAC address manually.

**Before you begin**

Ensure that you have enabled the vPC feature.

**SUMMARY STEPS**

1. configure terminal
2. vpc domain domain-id [shut | no shut]
3. system-mac mac-address
4. exit
5. show vpc role
6. copy running-config startup-config

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
| Example: | switch# configure terminal  
switch(config)# | |
| Step 2 | vpc domain domain-id [shut | no shut] | Enters the vPC domain number that you want to configure. The system enters vpc-domain configuration mode. |
| Example: | switch(config)# vpc domain 5  
switch(config-vpc-domain)# | |
| Step 3 | system-mac mac-address | Enters the MAC address that you want for the specified vPC domain in the following format: aaaa.bbbb.cccc. |
| Example: | switch(config-vpc-domain)# system-mac 23fb.4ab5.4c4e  
switch(config-vpc-domain)# | |
| Step 4 | exit | Exits vpc-domain configuration mode. |
| Example: | switch(config-vpc-domain)# exit  
switch# | |
| Step 5 | show vpc role | (Optional) Displays the vPC system MAC address. |
| Example: | switch# show vpc brief | |
Manually Configuring the System Priority

When you create a vPC domain, the system automatically creates a vPC system priority. However, you can also manually configure a system priority for the vPC domain.

**Note**

We recommend that you manually configure the vPC system priority when you are running LACP to ensure that the vPC peer devices are the primary devices on LACP. When you manually configure the system priority, ensure that you configure the same priority value on both vPC peer devices. If these values do not match, vPC does not come up.

**Before you begin**

Ensure that you have enabled the vPC feature.

**SUMMARY STEPS**

1. `configure terminal`
2. `vpc domain domain-id [shut | no shut]`
3. `system-priority priority`
4. `exit`
5. `show vpc role`
6. `copy running-config startup-config`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>Example: <code>switch# configure terminal</code> <code>switch(config)#</code></td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

#### Command or Action

| Step 2 | vpc domain domain-id [shut | no shut] |
|---|---|
| **Example:** | switch(config)# vpc domain 5 |
| | switch(config-vpc-domain)# |

Enters the vPC domain number that you want to configure. The system enters vpc-domain configuration mode.

<table>
<thead>
<tr>
<th>Step 3</th>
<th>system-priority priority</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>switch(config-vpc-domain)# system-priority 4000</td>
</tr>
<tr>
<td></td>
<td>switch(config-vpc-domain)#</td>
</tr>
</tbody>
</table>

Enters the system priority that you want for the specified vPC domain. The range of values is from 1 to 65535. The default value is 32667.

<table>
<thead>
<tr>
<th>Step 4</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>switch(config-vpc-domain)# exit</td>
</tr>
<tr>
<td></td>
<td>switch#</td>
</tr>
</tbody>
</table>

Exits vpc-domain configuration mode.

<table>
<thead>
<tr>
<th>Step 5</th>
<th>show vpc role</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>switch# show vpc role</td>
</tr>
</tbody>
</table>

(Optional) Displays the vPC system priority.

<table>
<thead>
<tr>
<th>Step 6</th>
<th>copy running-config startup-config</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>switch# copy running-config startup-config</td>
</tr>
</tbody>
</table>

(Optional) Copies the running configuration to the startup configuration.

---

**Example**

This example shows how to manually configure the vPC domain system priority:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# system-priority 4000
switch(config-vpc-domain)# exit
switch(config)#
```

---

**Manually Configuring the vPC Peer Device Role**

By default, the Cisco NX-OS software elects a primary and secondary vPC peer device after you configure the vPC domain and both sides of the vPC peer link. However, you might want to elect a specific vPC peer device as the primary device for the vPC. Then, you would manually configure the role value for the vPC peer device that you want as the primary device to be lower than the other vPC peer device.

vPCs do not support role preemption. If the primary vPC peer device fails, the secondary vPC peer device takes over to become operationally the vPC primary device. However, the original operational roles are not restored if the formerly primary vPC comes up again.

**Before you begin**

Ensure that you have enabled the vPC feature.
### SUMMARY STEPS

1. `configure terminal`
2. `vpc domain domain-id [shut | no shut]`
3. `role priority priority`
4. `exit`
5. `show vpc role`
6. `copy running-config startup-config`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config)#</code></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>`vpc domain domain-id [shut</td>
<td>no shut]`</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config)# vpc domain 5</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-vpc-domain)#</code></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><code>role priority priority</code></td>
<td>Enters the role priority that you want for the vPC system priority. The range of values is from 1 to 65636, and the default value is 32667. A lower value means that this switch has a better chance of being the primary vPC.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-vpc-domain)# role priority 4</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-vpc-domain)#</code></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><code>exit</code></td>
<td>Exits vpc-domain configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config)# exit</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch#</code></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><code>show vpc role</code></td>
<td>(Optional) Displays the vPC system priority.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch# show vpc role</code></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td><code>copy running-config startup-config</code></td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch# copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>

### Example

This example shows how to manually configure the role priority of the vPC peer device:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# role priority 4
```
Configuring the Tracking Feature on a Single-Module vPC

If you must configure all the vPC peer links and core-facing interfaces on a single module, you should configure a track object and a track list that is associated with the Layer 3 link to the core and on all the links on the vPC peer link on both primary vPC peer devices. Once you configure this feature and if the primary vPC peer device fails, the system automatically suspends all the vPC links on the primary vPC peer device. This action forces all the vPC traffic to the secondary vPC peer device until the system stabilizes.

You must put this configuration on both vPC peer devices. Additionally, you should put the identical configuration on both vPC peer devices because either device can become the operationally primary vPC peer device.

Before you begin

Ensure that you have enabled the vPC feature.

Ensure that you have configured the track object and the track list. Ensure that you assign all interfaces that connect to the core and to the vPC peer link to the track-list object on both vPC peer devices.

SUMMARY STEPS

1. configure terminal
2. vpc domain domain-id [shut | no shut]
3. track track-object-id
4. exit
5. show vpc brief
6. copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> vpc domain domain-id [shut</td>
<td>no shut]</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# vpc domain 5</td>
<td></td>
</tr>
<tr>
<td>switch(config-vpc-domain)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> track track-object-id</td>
<td>Adds the previously configured track-list object with its</td>
</tr>
<tr>
<td>Example:</td>
<td>associated interfaces to the vPC domain. See the Cisco</td>
</tr>
<tr>
<td>switch(config-vpc-domain)# track object 23</td>
<td>Nexus 9000 Series NX-OS Unicast Routing Configuration Guide</td>
</tr>
<tr>
<td>switch(config-vpc-domain)#</td>
<td>for information about configuring object tracking and track lists.</td>
</tr>
</tbody>
</table>
## Configuring for Recovery After an Outage

If an outage occurs, the vPC waits for a peer adjacency to form on a switch reload. This situation can result in an unacceptably long service disruption. You can configure the Cisco Nexus 9000 Series device to restore vPC services when its peer fails to come on line.

### Configuring Reload Restore

The `reload restore` command and procedure described in this section is deprecated. We recommend that you use the `auto-recovery` command and procedure described in the “Configuring an Autorecovery” section.

You can configure the Cisco Nexus 9000 Series device to restore vPC services when its peer fails to come online by using the `reload restore` command.

**Before you begin**

Ensure that you have enabled the vPC feature.

**SUMMARY STEPS**

1. `configure terminal`
2. `vpc domain domain-id [shut | no shut]`
3. `reload restore [delay time-out]`
4. `exit`

---

### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>exit</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>switch(config-vpc-domain)# exit</code></td>
</tr>
<tr>
<td></td>
<td><code>switch#</code></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>show vpc brief</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>switch# show vpc brief</code></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>copy running-config startup-config</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>switch# copy running-config startup-config</code></td>
</tr>
</tbody>
</table>
5. `show running-config vpc`
6. `show vpc consistency-parameters interface port-channel number`
7. `copy running-config startup-config`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>switch# <code>configure terminal</code></td>
<td>switch(config)#</td>
</tr>
<tr>
<td>Step 2</td>
<td>`vpc domain domain-id [shut</td>
<td>no shut]`</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>switch(config)# <code>vpc domain 5</code></td>
<td>switch(config-vpc-domain)#</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>reload restore [delay time-out]</code></td>
<td>Configures the vPC to assume its peer is not functional and to bring up the vPC. The default delay is 240 seconds. You can configure a time-out delay from 240 to 3600 seconds. Use the <code>no</code> form of the command to reset the vPC to its default settings.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>switch(config-vpc-domain)# <code>reload restore</code></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>exit</code></td>
<td>Exits vpc-domain configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>switch(config-vpc-domain)# <code>exit</code></td>
<td>switch#</td>
</tr>
<tr>
<td>Step 5</td>
<td><code>show running-config vpc</code></td>
<td>(Optional) Displays information about the vPC, specifically the reload status.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>switch# <code>show running-config vpc</code></td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td><code>show vpc consistency-parameters interface port-channel number</code></td>
<td>(Optional) Displays information about the vPC consistency parameters for the specified interface.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>switch# <code>show vpc consistency-parameters interface port-channel 1</code></td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td><code>copy running-config startup-config</code></td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>switch# <code>copy running-config startup-config</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td>To ensure the reload feature is enabled, you should perform this step.</td>
</tr>
</tbody>
</table>
Example

This example shows how to set the vPC reload restore feature and save it in the switch startup configuration:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# vpc domain 5
switch(config-vpc-domain)# reload restore

Warning:
Enables restoring of vPCs in a peer-detached state after reload, will wait for 240 seconds (by default) to determine if peer is un-reachable

switch(config-vpc-domain)# exit
switch(config)# exit
switch# copy running-config startup-config
```

```
!Command: show running-config vpc
!Time: Wed Mar 24 18:43:54 2010
version 5.0(2)
feature vpc
logging level vpc 6
vpc domain 5
reload restore

This example shows how to examine the consistency parameters:

```
switch# show vpc consistency-parameters interface port-channel 1

Legend: 
Type 1 : vPC will be suspended in case of mismatch 
Name Type Local Value Peer Value
---------- ---- ----------- ---------------
STP Port Type 1 Default -
STP Port Guard 1 None -
STP MST Simulate PVST 1 Default -
mode 1 on -
Speed 1 1000 Mb/s -
Duplex 1 full -
Port Mode 1 trunk -
Native Vlan 1 1 -
MTU 1 1500 -
Allowed VLANs - 1-3967,4048-4093
Local suspended VLANs
```

Configuring an Autorecovery

You can configure the Cisco Nexus 9000 Series device to restore vPC services when its peer fails to come online by using the auto-recovery command.

Note

The auto-recovery feature is not enabled by default on Cisco Nexus 9000 Switches. When the object tracking is triggered, the vPC secondary peer device does not change its role to that primary device and it reinitializes the vPC legs. You must manually configure auto-recovery on the vPC secondary peer device so that it can take over the primary role and reinitialize its vPC legs.
Before you begin

Ensure that you have enabled the vPC feature.

### SUMMARY STEPS

1. configure terminal
2. vpc domain domain-id [shut | no shut]
3. auto-recovery [reload-delay time]
4. exit
5. show running-config vpc
6. show vpc consistency-parameters interface port-channel number
7. copy running-config startup-config

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> vpc domain domain-id [shut</td>
<td>no shut]</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# vpc domain 5</td>
<td></td>
</tr>
<tr>
<td>switch(config-vpc-domain)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> auto-recovery [reload-delay time]</td>
<td>Configures the vPC to assume its peer is not functional and to bring up the vPC, and specifies the time to wait after a reload to restore the vPC. The default delay is 240 seconds. You can configure a delay from 240 to 3600 seconds. Use the no form of the command to reset the vPC to its default settings.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-vpc-domain)# auto-recovery</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> exit</td>
<td>Exits vpc-domain configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-vpc-domain)# exit</td>
<td></td>
</tr>
<tr>
<td>switch#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> show running-config vpc</td>
<td>(Optional) Displays information about the vPC, specifically the reload status.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# show running-config vpc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> show vpc consistency-parameters interface port-channel number</td>
<td>(Optional) Displays information about the vPC consistency parameters for the specified interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# show vpc consistency-parameters</td>
<td></td>
</tr>
<tr>
<td>interface port-channel 1</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring vPCs

#### Configuring the Suspension of Orphan Ports

When a device that is not vPC-capable connects to each peer, the connected ports are known as orphan ports because they are not members of a vPC. You can explicitly declare physical interfaces as orphan ports to be suspended (shut down) by the secondary peer when it suspends its vPC ports in response to a peer link or peer-keepalive failure. The orphan ports are restored when the vPC is restored.

**Note**

You can configure vPC orphan port suspension only on physical ports, portchannels. However, you cannot configure the same on individual port channel member ports.

**Before you begin**

Ensure that you have enabled the vPC feature.

**SUMMARY STEPS**

1. configure terminal
2. show vpc orphan-ports
3. interface type slot/port
4. vpc orphan-port suspend
5. exit
6. copy running-config startup-config

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>copy running-config startup-config</td>
<td>(Optional) Copies the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>

**Example**

This example shows how to set the vPC autorecovery feature and save it in the switch startup configuration:

```plaintext
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# vpc domain 5
switch(config-vpc-domain)# auto-recovery
Warning:
Enables restoring of vPCs in a peer-detached state after reload, will wait for 240 seconds to determine if peer is un-reachable
switch(config-vpc-domain)# exit
switch(config)# exit
switch# copy running-config startup-config
```
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
| Example: | switch# configure terminal  
switch(config)# | |
| Step 2 | show vpc orphan-ports | (Optional) Displays a list of the orphan ports. |
| Example: | switch# show vpc orphan-ports | |
| Step 3 | interface type slot/port | Specifies an interface to configure, and enters interface configuration mode. |
| Example: | switch(config)# interface ethernet 3/1  
switch(config-if)# | |
| Step 4 | vpc orphan-port suspend | Configures the selected interface as a vPC orphan port to be suspended by the secondary peer in the case of a vPC failure. |
| Example: | switch(config-if)# vpc orphan-ports suspend  
switch(config-if)# exit  
switch(config)# | |
| Step 5 | exit | Exits interface configuration mode. |
| Example: | switch(config-if)# exit  
switch# | |
| Step 6 | copy running-config startup-config | (Optional) Copies the running configuration to the startup configuration. |
| Example: | switch# copy running-config startup-config | |

### Example

This example shows how to configure an interface as a vPC orphan port to be suspended by the secondary peer in the case of a vPC failure:

```
switch# configure terminal  
switch(config)# interface ethernet 3/1  
switch(config-if)# vpc orphan-ports suspend  
switch(config-if)# exit  
switch(config)#
```

Beginning Cisco NX-OS Release 9.2(1), the output of the `show vpc orphan-ports` command is slightly different from that of the earlier releases. This example shows the output of `show vpc orphan-ports` command:

```
switch# show vpc orphan-ports  
-------:Going through port database. Please be patient.:-------  
VLAN Orphan Ports  
------- -------------------------  
1 Eth1/18, Eth3/23
```
Configuring Delay Restore on an Orphan Port

You can configure `delay restore orphan-port` command on Cisco Nexus 9000 Series switches to configure a restore timer that delays the bringing up of restored device's orphan port.

**Note**
The delay restore orphan-port command applies only to interfaces that has vpc orphan-port suspend command configured. Other orphan ports may not delay bringing up devices.

**SUMMARY STEPS**

1. configure terminal
2. switch(config) # vpc domain <domain>
3. switch(config) # peer-switch
4. switch(config) # show vpc peer-keepalive
5. switch(config) # delay restore { time }
6. switch(config) # peer-gateway
7. switch(config) # delay restore orphan-port

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2 switch(config) # vpc domain &lt;domain&gt;</td>
<td>Configure the VPC domain number.</td>
</tr>
<tr>
<td>Step 3 switch(config) # peer-switch</td>
<td>Define the peer switch.</td>
</tr>
<tr>
<td>Step 4 switch(config) # show vpc peer-keepalive</td>
<td>Displays information about the peer keepalive messages.</td>
</tr>
<tr>
<td>Step 5 switch(config) # delay restore { time }</td>
<td>Number of seconds to delay bringing up the restored vPC peer device. The range is from 1 to 3600.</td>
</tr>
<tr>
<td>Step 6 switch(config) # peer-gateway</td>
<td>To enable Layer 3 forwarding for packets destined to the gateway MAC address of the virtual Port Channel (vPC), use the peer-gateway command. To disable Layer 3 forwarding packets, use the no form of this command.</td>
</tr>
<tr>
<td>Step 7 switch(config) # delay restore orphan-port</td>
<td>Number of seconds to delay bringing up the restored device's orphan port</td>
</tr>
</tbody>
</table>
Configuring the vPC Peer Switch

You can configure the Cisco Nexus 9000 Series device to make a pair of vPC devices appear as a single STP root in the Layer 2 topology.

Configuring a Pure vPC Peer Switch Topology

You can configure a pure vPC peer switch topology by using the peer-switch command and then setting the best possible (lowest) spanning tree bridge priority value.

**Before you begin**

Ensure that you have enabled the vPC feature.

**Note**

When using a non-VPC dedicated trunk link between the VPC peers, the non-VPC VLANs should have a different global priority on the peers to prevent STP from blocking the VLANs.

**SUMMARY STEPS**

1. `configure terminal`
2. `vpc domain domain-id [shut | no shut]`
3. `peer-switch`
4. `spanning-tree vlan vlan-range priority value`
5. `exit`
6. `show spanning-tree summary`
7. `copy running-config startup-config`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: switch# configure terminal switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> vpc domain domain-id [shut</td>
<td>no shut]</td>
</tr>
<tr>
<td>Example: switch(config)# vpc domain 5 switch(config-vpc-domain)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> peer-switch</td>
<td>Enables the vPC switch pair to appear as a single STP root in the Layer 2 topology. Use the no form of the command to disable the peer switch vPC topology.</td>
</tr>
<tr>
<td>Example: switch(config-vpc-domain)# peer-switch</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> spanning-tree vlan vlan-range priority value</td>
<td>Configures the bridge priority of the VLAN. Valid values are multiples of 4096. The default value is 32768.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring vPCs

#### Configuring a Hybrid vPC Peer Switch Topology

You can configure a hybrid vPC and non-vPC peer switch topology by using the spanning-tree pseudo-information command to change the designated bridge ID so that it meets the STP VLAN-based load-balancing criteria and then change the root bridge ID priority to a value that is better than the best bridge priority. You then enable the peer switch.

**Before you begin**

Ensure that you have enabled the vPC feature.

When using a non-VPC dedicated trunk link between the VPC peers, the non-VPC VLANs should have a different pseudo root priority on the peers to prevent STP from blocking the VLANs.

**SUMMARY STEPS**

1. configure terminal
2. spanning-tree pseudo-information

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>switch(config)# spanning-tree vlan 1 priority 8192</code></td>
<td>Exits vpc-domain configuration mode.</td>
</tr>
<tr>
<td><strong>Step 5</strong> exit</td>
<td>(Optional) Displays a summary of the spanning tree port states including the vPC peer switch.</td>
</tr>
</tbody>
</table>
| **Example:** `switch(config-vpc-domain)# exit
switch#` | |
| **Step 6** show spanning-tree summary | (Optional) Copies the running configuration to the startup configuration. |
| **Example:** `switch# show spanning-tree summary` | |
| **Step 7** copy running-config startup-config | |
| **Example:** `switch# copy running-config startup-config` | |
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> spanning-tree pseudo-information</td>
<td>Configures the spanning tree pseudo information.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch(config)# spanning-tree pseudo-information</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config-pseudo)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> vlan <em>vlan-id</em> designated priority <em>priority</em></td>
<td>Configures the designated bridge priority of the VLAN. Valid values are multiples of 4096 from 0 to 61440.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch(config-pseudo)# vlan 1 designated priority 8192</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> vlan <em>vlan-id</em> root priority <em>priority</em></td>
<td>Configures the root bridge priority of the VLAN. Valid values are multiples of 4096 from 0 to 61440.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch(config-pseudo)# vlan 1 root priority 4096</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> vpc domain <em>domain-id</em> [shut</td>
<td>no shut]</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch(config)# vpc domain 5</code></td>
<td></td>
</tr>
<tr>
<td><code>switch(config-vpc-domain)#</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> peer-switch</td>
<td>Enables the vPC switch pair to appear as a single STP root in the Layer 2 topology. Use the no form of the command to disable the peer switch vPC topology.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch(config-vpc-domain)# peer-switch</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> exit</td>
<td>Exits vpc-domain configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>switch(config-vpc-domain)# exit</code></td>
<td></td>
</tr>
<tr>
<td><code>switch#</code></td>
<td></td>
</tr>
</tbody>
</table>
Configuring Hitless vPC Role Change

Complete these steps to enable hitless vPC role change.

**Before you begin**

- Ensure that the vPC feature is enabled.
- Ensure that the vPC peer link is up
- Verify the role priority of devices

**SUMMARY STEPS**

1. vpc role preempt
2. show vpc role

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 vpc role preempt Example: switch# vpc role preempt switch(config)#</td>
<td>Enable hitless vPC role change.</td>
</tr>
</tbody>
</table>

---

Example

This example shows how to configure a hybrid vPC peer switch topology:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# spanning-tree pseudo-information
switch(config-pseudo)# vlan 1 designated priority 8192
switch(config-pseudo)# vlan 1 root priority 4096
switch(config-pseudo)# vpc domain 5
switch(config-vpc-domain)# peer-switch
switch(config-vpc-domain)# exit
switch(config)#
```
Use Case Scenario for vPC Role Change

The hitless vPC role change feature can be used in the following scenarios:

• **Role change request**—When you want to change the roles of the peer devices in a vPC domain.

• **Primary switch reload**—When the devices come up after a reload and roles are defined, you can use the hitless vPC role change feature to restore the roles. For example, after a reload if the primary device takes the role of operational secondary and the secondary device takes the role of primary operational, you can change the vPC peer roles to their original defined roles using the `vpc role preempt` command.

---

**Note**

Always check the existing device role priority before switching vPC role.

• **Dual-active recovery**—In a dual-active recovery scenario, the vPC primary switch continues to be (operational) primary, but the vPC secondary switch becomes the targeted primary switch and keeps its vPC member ports up. You can use the vPC hitless feature and restore the device roles. After the
Dual-active recovery, if one side is operational primary and the other side operational secondary, then you can use the `vpc role preempt` command to restore the device roles to be primary and secondary

### Enabling STP to Use the Cisco MAC Address

This procedure enables STP to use the Cisco MAC address (00:26:0b:xx:xx:xx).

**Before you begin**

Ensure that you have enabled the vPC feature.

### SUMMARY STEPS

1. `configure terminal`
2. `vpc domain domain-id`
3. `[no] mac-address bpdu source version 2`
4. `exit`
5. (Optional) `copy running-config startup-config`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>Example: <code>switch# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Creates a vPC domain if it does not already exist, and enters vpc-domain configuration mode.</td>
</tr>
<tr>
<td><code>vpc domain domain-id</code></td>
<td></td>
</tr>
<tr>
<td>Example: <code>switch(config)# vpc domain 5</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enables STP to use the Cisco MAC address (00:26:0b:xx:xx:xx) as the source address of BPDUs generated on vPC ports.</td>
</tr>
<tr>
<td><code>[no] mac-address bpdu source version 2</code></td>
<td></td>
</tr>
<tr>
<td>Example: <code>switch(config-vpc-domain)# mac-address bpdu source version 2</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Exits vpc-domain configuration mode.</td>
</tr>
<tr>
<td><code>exit</code></td>
<td></td>
</tr>
<tr>
<td>Example: <code>switch(config-vpc-domain)# exit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td>(Optional) <code>copy running-config startup-config</code></td>
<td></td>
</tr>
<tr>
<td>Example: <code>switch(config)# copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>
Verifying the vPC Configuration

To display vPC configuration information, perform one of the following tasks:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>show feature</td>
<td>Displays whether the vPC is enabled or not.</td>
</tr>
<tr>
<td>show vpc brief</td>
<td>Displays brief information about the vPCs.</td>
</tr>
<tr>
<td>show vpc consistency-parameters</td>
<td>Displays the status of those parameters that must be consistent across all vPC interfaces.</td>
</tr>
<tr>
<td>show running-config vpc</td>
<td>Displays running configuration information for vPCs.</td>
</tr>
<tr>
<td>show port-channel capacity</td>
<td>Displays how many port channels are configured and how many are still available on the device.</td>
</tr>
<tr>
<td>show vpc statistics</td>
<td>Displays statistics about the vPCs.</td>
</tr>
<tr>
<td>show vpc peer-keepalive</td>
<td>Displays information about the peer-keepalive messages.</td>
</tr>
<tr>
<td>show vpc role</td>
<td>Displays the peer status, the role of the local device, the vPC system MAC address and system priority, and the MAC address and priority for the local vPC device.</td>
</tr>
</tbody>
</table>

Monitoring vPCs

Use the **show vpc statistics** command to display vPC statistics.

**Note**
This command displays the vPC statistics only for the vPC peer device that you are working on.

Configuration Examples for vPCs

The following example shows how to configure vPC on device A as shown in the figure:
1. Enable vPC and LACP.

switch# configure terminal
switch(config)# feature vPC
switch(config)# feature lacp

2. (Optional) Configure one of the interfaces that you want to be a peer link in the dedicated port mode.

switch(config)# interface ethernet 7/1, ethernet 7/3, ethernet 7/5, ethernet 7/7
switch(config-if)# shutdown
switch(config-if)# exit
switch(config)# interface ethernet 7/1
switch(config-if)# rate-mode dedicated
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)#

3. (Optional) Configure the second, redundant interface that you want to be a peer link in the dedicated port mode.

switch(config)# interface ethernet 7/2, ethernet 7/4, ethernet 7/6, ethernet 7/8
switch(config-if)# shutdown
switch(config-if)# exit
switch(config)# interface ethernet 7/2
switch(config-if)# rate-mode dedicated
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)#

4. Configure the two interfaces (for redundancy) that you want to be in the peer link to be an active Layer 2 LACP port channel.

switch(config)# interface ethernet 7/1-2
switch(config-if)# switchport
switch(config-if)# switchport mode trunk
5. Create and enable the VLANs.

switch(config)# vlan 1-50
switch(config-vlan)# no shutdown
switch(config-vlan)# exit

6. Create a separate VRF for the vPC peer-keepalive link and add a Layer 3 interface to that VRF.

switch(config)# vrf context pkal
switch(config-vrf)# exit
switch(config)# interface ethernet 8/1
switch(config-if)# vrf member pkal
switch(config-if)# ip address 172.23.145.218/24
switch(config-if)# no shutdown
switch(config-if)# exit

7. Create the vPC domain and add the vPC peer-keepalive link.

switch(config)# vpc domain 1
switch(config-vpc-domain)# peer-keepalive
destination 172.23.145.217 source 172.23.145.218 vrf pkal
switch(config-vpc-domain)# exit

8. Configure the vPC peer link.

switch(config)# interface port-channel 20
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk allowed vlan 1-50
switch(config-if)# vpc peer-link
switch(config-if)# exit
switch(config)#

9. Configure the interface for the port channel to the downstream device of the vPC.

switch(config)# interface ethernet 7/9
switch(config-if)# switchport mode trunk
switch(config-if)# allowed vlan 1-50
switch(config-if)# native vlan 20
switch(config-if)# channel-group 50 mode active
switch(config-if)# exit
switch(config)# interface port-channel 50
switch(config-if)# vpc 50
switch(config-if)# exit
switch(config)#

10. Save the configuration.

switch(config)# copy running-config startup-config

---

**Note**

If you configure the port channel first, ensure that it is a Layer 2 port channel.
## Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Related Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>System management</td>
<td>System management</td>
</tr>
<tr>
<td>High availability</td>
<td>High availability</td>
</tr>
<tr>
<td>Release Notes</td>
<td>Release Notes</td>
</tr>
</tbody>
</table>
Configuring IP Tunnels

Information About IP Tunnels

IP tunnels can encapsulate a same-layer or higher layer protocol and transport the result over IP through a tunnel created between two devices.

IP Tunnel Overview

IP tunnels consist of the following three main components:

- Passenger protocol—The protocol that needs to be encapsulated. IPv4 is an example of a passenger protocol.
- Carrier protocol—The protocol that is used to encapsulate the passenger protocol. Cisco NX-OS supports GRE as a carrier protocol.
- Transport protocol—The protocol that is used to carry the encapsulated protocol. IPv4 is an example of a transport protocol. An IP tunnel takes a passenger protocol, such as IPv4, and encapsulates that protocol within a carrier protocol, such as GRE. The device then transmits this carrier protocol over a transport protocol, such as IPv4.

You configure a tunnel interface with matching characteristics on each end of the tunnel.

You must enable the tunnel feature before you can configure it. The system automatically takes a checkpoint prior to disabling the feature, and you can roll back to this checkpoint. See the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide for information about rollbacks and checkpoints.
**GRE Tunnels**

You can use generic routing encapsulation (GRE) as the carrier protocol for a variety of passenger protocols.

The following figure shows the IP tunnel components for a GRE tunnel. The original passenger protocol packet becomes the GRE payload and the device adds a GRE header to the packet. The device then adds the transport protocol header to the packet and transmits it.

*Figure 32: GRE PDU*

---

**Point-to-Point IP-in-IP Tunnel Encapsulation and Decapsulation**

The point-to-point IP-in-IP encapsulation and decapsulation is a type of tunnel that you can create to send encapsulated packets from a source tunnel interface to a destination tunnel interface. This type of tunnel will carry both inbound and outbound traffic.

---

**Note**

The selection of IP-in-IP tunnel based on the PBR policy is not supported.

**Note**


---

**Multi-Point IP-in-IP Tunnel Decapsulation**

The multi-point IP-in-IP decapsulate-any is a type of tunnel that you can create to decapsulate packets from any number of IP-in-IP tunnels to one tunnel interface. This tunnel will not carry any outbound traffic. However, any number of remote tunnel endpoints can use a tunnel configured this way as their destination.

---

**Path MTU Discovery**

Path maximum transmission unit (MTU) discovery (PMTUD) prevents fragmentation in the path between two endpoints by dynamically determining the lowest MTU along the path from the packet's source to its destination. PMTUD reduces the send MTU value for the connection if the interface receives information that the packet would require fragmentation.
When you enable PMTUD, the interface sets the Don't Fragment (DF) bit on all packets that traverse the tunnel. If a packet that enters the tunnel encounters a link with a smaller MTU than the MTU value for the packet, the remote link drops the packet and sends an ICMP message back to the sender of the packet. This message indicates that fragmentation was required (but not permitted) and provides the MTU of the link that dropped the packet.

Note

PMTUD on a tunnel interface requires that the tunnel endpoint can receive ICMP messages generated by devices in the path of the tunnel. Check that ICMP messages can be received before using PMTUD over firewall connections.

High Availability

IP tunnels support stateful restarts. A stateful restart occurs on a supervisor switchover. After the switchover, Cisco NX-OS applies the runtime configuration after the switchover.

Licensing Requirements for IP Tunnels

The following table shows the licensing requirements for this feature:

<table>
<thead>
<tr>
<th>Product</th>
<th>License Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco NX-OS</td>
<td>IP tunnels require an Enterprise Services license. For a complete explanation of the</td>
</tr>
<tr>
<td></td>
<td>Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the Cisco</td>
</tr>
<tr>
<td></td>
<td>NX-OS Licensing Guide.</td>
</tr>
</tbody>
</table>

Prerequisites for IP Tunnels

IP tunnels have the following prerequisites:

- You must be familiar with TCP/IP fundamentals to configure IP tunnels.
- You are logged on to the switch.
- You must enable the tunneling feature in a device before you can configure and enable any IP tunnels.

Guidelines and Limitations

IP tunnels have the following configuration guidelines and limitations:

- Guidelines for `source-direct` and `ipv6ipv6-decapsulate-any` options for tunnels:
  - The `source-direct` command is supported on the Cisco Nexus 9500 platform switches with the Application Spine Engine (ASE) and the Leaf Spine Engine (LSE).
Cisco Nexus 9500 platform switches with the Network Forwarding Engine (NFE) do not support the `tunnel source direct` command.

The `tunnel source direct` command with the `tunnel mode ipv6ipv6 decapsulate-any` command on the Cisco Nexus 9500 platform switches is only supported in the MPLS heavy routing template.

- The IP tunnel supports the `tunnel source` CLI command with interface, IPv4 address, IPv6 address, or IPv4 prefix. You can configure IP-in-IP tunnel decapsulation on directly connected IP addresses (for example, physical interface, port-channel, loopback, and SVI) using the new `tunnel source direct` CLI command. You can select the IP ECMP links when there are multiple IP links between the two switches. A single tunnel interface can decapsulate the tunneled packets whose outer destination IP is any of the IPv4 or IPv6 address that is locally configured and it is operationally `Up` in the switch.

- Currently, `tunnel mode ipip decapsulate-any` is supported for decapsulating IPv4 payload over IPv4 transport (IPv4inIPv4 packets). `tunnel mode ipv6ipv6 decapsulate-any` command is introduced to support IPv6 payload over IPv6 transport (IPv6inIPv6 packets).

- The `tunnel source direct` and `tunnel mode ipv6ipv6 decapsulate-any` CLI commands are not supported on Cisco Nexus 9500 platform switches with the Network Formation Engine (NFE).

- The `tunnel source direct` CLI command is supported only when an administrator uses the IP-in-IP decapsulation to source route the packets through the network. The source-direct tunnel is always operationally `Up` unless it is administratively shut down. The directly connected interfaces are identified using the `show ip route direct` CLI command.

- The `tunnel source direct` CLI command is supported only on decapsulate-any tunnel modes, for example, `tunnel mode ipip decapsulate-any` and `tunnel mode ipv6ipv6 decapsulate-any`.

- Auto-recovery for source-direct is not supported.

- For `ipv6ipv6 decapsulate-any`, inter-VRF is not supported. The tunnel interface VRF (iVRF) and tunnel transport or forwarding VRF (fVRF) must be the same. Only one decapsulate-any tunnel (irrespective of VRF) can be present in Cisco Nexus 9200 and 9300-EX Series switches.

- To enable IPv6 on ipv6ipv6 decap-any tunnel interface, configure a valid IPv6 address or configure `ipv6 address use-link-local-only` under the tunnel interface.

- See the following hardware limitations on the maximum sources that can be accommodated on a source direct tunnel and the related behavior:

  - Source direct tunnel is now supported for Cisco Nexus 9000 Series switches with Network Forwarding Engine (NFE), Application Spine Engine (ASE), and Leaf Spine Engine (LSE). Most of the limitations are only in case of scaled SIP (number of total IP/IPv6 addresses on the interfaces (L3, sub-interface, PC, PC-sub interfaces, loopback, SVI, and any secondary IP/IPv6 addresses.)

See the following sample use cases.

- Use Case 1: Non-deterministic behavior of which SIP gets installed if the number of IP/IPv6 interface scale is more.

  Both the switches have 512 entries for tunnel SIP. With tunnel source, direct any IP or IPv6 address w.r.t `ipip or ipv6ipv6 decap any` with tunnel source gets installed in the above table. The insertion of these entries is on a first come first serve basis without any CLI command to control which interface IP addresses get installed. If the system has more number of IP/IPv6.
interfaces to be installed, the behavior is non-deterministic (The behavior can change across reload with interface flaps.)

- Use Case 2: The scale numbers are different in both switches and each has its own advantages and disadvantages.

IPv4 individual scale can be more (up to 512) in case of switches with NFE but it is shared with IPv6. In the switches with ASE and LSE, the IPv4 individual scale can be 256 but it is not shared with IPv6.

Whenever the tunnel decap table gets filled, the TABLE_FULL error is displayed. If some entry gets deleted after the table gets full, the table full error is cleared.

**Table 14: Scale Numbers**

<table>
<thead>
<tr>
<th>Commands</th>
<th>Switches with NFE: Table size 512, v4 takes 1 entry, v6 takes 4 entries</th>
<th>Switches with ASE and LSE: Table size 512, v4 takes 1 entry, v6 takes 2 entries (paired index)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPip decap any with tunnel source direct</td>
<td>Shared between v4 and v6, v6 takes 4 entries v4 + 4 *v6 =512 Maximum entries can be 512 with no v6 entries</td>
<td>Dedicated 256</td>
</tr>
<tr>
<td>IPv6IPv6 decap any with tunnel source direct</td>
<td>Shared between v4 and v6, v6 takes 4 entries v4 + 4 *v6 =512 Maximum entries can be 128 with no v4 entries</td>
<td>Dedicated 128</td>
</tr>
</tbody>
</table>

- Use Case 3: Auto-recovery is not supported.

If any entry does not get installed in the hardware due to exhaustion of above table, removal of an already installed IP/IPv6 from interfaces does not automatically trigger the addition of the failed SIP in the table though the table has space now. You need to flap the tunnel interface or IP interface to get them installed.

However, if an entry does not get installed in the hardware due to a duplicate entry (if there was already a **decap-any** with one source present and now the **source direct tunnel** CLI command is configured, there is a duplicate entry for the prior source configured) that was taken care of by removing the entry only when both the tunnels get deleted.

- For Cisco Nexus 9000 Series switches with Network Forwarding Engine (NFE) and Application Spine Engine (ASE), the syslog is different as the dedicated IPv4 and IPv6 decap antries are carved in the syslog. If the **tunnel-decap-table** is full, the user gets a syslog as follows:

```
2017 Apr 6 12:18:04 switch %$ VDC-1 %$ %IPFIB-2-FIB_HW_IPV4_TUNNEL_DECAP_TABLE_FULL: IPv4 tunnel decap hardware table full.
IP tunnel decapsulation may not work for some GRE/IPinIP traffic
```

```
2017 Apr 6 12:18:11 switch %$ VDC-1 %$ %IPFIB-2-FIB_HW_IPV6_TUNNEL_DECAP_TABLE_FULL:
```
IPv6 tunnel decap hardware table full. 
IP tunnel decapsulation may not work for some GRE/IPinIP traffic

If the table is full and once some entry gets deleted from the table (due to an interface being operationally down or removal of IP address), the clear syslog for the table is displayed. Note that the deletion of the tunnel removes all the entries that are added as part of that tunnel.

2017 Apr 5 13:29:25 switch %$ VDC-1 %$ %IPFIB-2-FIB_HW_IPV4_TUNNEL_DECAP_TABLE_FULL_CLRD: IPv4 tunnel decap hardware table full exception cleared

2017 Apr 4 19:41:22 switch %$ VDC-1 %$ %IPFIB-2-FIB_HW_IPV6_TUNNEL_DECAP_TABLE_FULL_CLRD: IPv6 tunnel decap hardware table full exception cleared

- IP-in-IP tunnel decapsulation is supported on IPv6 enabled networks.

```
! interface tunnel 1
ipv6 address use-link-local-only <<< enable IPv6
tunnel mode ipv6ipv6 decapsulate-any
tunnel source direct
description IPinIP Decapsulation Interface
mtu 1476
no shutdown
```

- The `show` commands with the `internal` keyword are not supported.

- Cisco NX-OS supports only the following protocols:
  - IPv4 passenger protocol.
  - GRE carrier protocol.

- Cisco NX-OS supports the following maximum number of tunnels:
  - IP tunnels - 8 tunnels.
  - GRE and IP-in-IP regular tunnels - 8 tunnels.

- IP tunnels do not support access control lists (ACLs) or QoS policies.

- Cisco NX-OS supports the GRE header defined in IETF RFC 2784. Cisco NX-OS does not support tunnel keys and other options from IETF RFC 1701.

- Cisco NX-OS does not support GRE tunnel keepalives.

- All unicast routing protocols are supported by IP tunnels.

- The IP tunnel interface cannot be configured to be a span source or destination.

- IP tunnels do not support PIM or other Multicast features and protocols.

- The selection of IP-in-IP tunnel based on the PBR policy is not supported.

- IP tunnels are supported only in the default `system routing` mode and not in other modes.

- When configuring a tunnel interface to `pip mode`, the maximum mtu value is 9196.
When downgrading from NX-OS 9.2(1) to an earlier release, with a tunnel interface in \textit{ipip mode} that has an mtu value of 9196, the mtu configuration is lost as a result of the downgrade operation. As a best practice, adjust the mtu value to 9192 before commencing the downgrade to avoid losing the mtu configuration.

- When configuring a tunnel interface to \textit{ipip mode}, the default mtu value is 1480.

When downgrading from NX-OS 9.2(1) or later release to an earlier release, with a tunnel interface in \textit{ipip mode} with no explicit mtu configuration, the mtu value changes as a result of the downgrade operation from 1480 to 1476. As a best practice, adjust the mtu value to 1476 before commencing the downgrade to avoid any changes to the mtu value.

When upgrading to NX-OS 9.2(1) or later release, with a tunnel interface in \textit{ipip mode} with no explicit mtu configuration, the mtu value changes as a result of the upgrade operation from 1476 to 1480. As a best practice, adjust the mtu value to 1480 before commencing the upgrade to avoid any changes to the mtu value.

- On Cisco Nexus 9200 Series switches, GRE packets that are received on an IP-in-IP tunnel are not dropped as expected and are instead forwarded to the packet destination.

- Tx packets originating from the switch, such as control pkts, are not included in Tx statistics.

- Tunnel destinations that are reachable over another tunnel are not supported.

- The consistency checker is not supported for routes over a tunnel.

- Non-IP routing protocols, such as isis, are not supported over IP-in-IP tunnels.

- RFC5549 is not supported over tunnels.

- BGP adjacency over tunnel is not supported in a scenario where the tunnel interface and tunnel source are in same VRF (example: VRF-A) and tunnel destination is reachable with route-leak from opposite end (example: via VRF-B)

### Default Settings

The following table lists the default settings for IP tunnel parameters.

\textit{Table 15: Default IP Tunnel Parameters}

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path MTU discovery age timer</td>
<td>10 minutes</td>
</tr>
<tr>
<td>Path MTU discovery minimum MTU</td>
<td>64</td>
</tr>
<tr>
<td>Tunnel feature</td>
<td>Disabled</td>
</tr>
</tbody>
</table>
Configuring IP Tunnels

If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Enabling Tunneling

You must enable the tunneling feature before you can configure any IP tunnels.

SUMMARY STEPS

1. configure terminal
2. feature tunnel
3. exit
4. show feature
5. copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>feature tunnel</td>
<td>Allows the creation of a new tunnel interface.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>To disable the tunnel interface feature, use the no form of this command.</td>
</tr>
<tr>
<td></td>
<td>switch(config)# feature tunnel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>exit</td>
<td>Exits the interface mode and returns to the configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# exit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch#</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>show feature</td>
<td>(Optional) Displays information about the features enabled on the device.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# show feature</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>copy running-config startup-config</td>
<td>(Optional) Saves this configuration change.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# copy running-config startup-config</td>
<td></td>
</tr>
</tbody>
</table>
Creating a Tunnel Interface

You can create a tunnel interface and then configure this logical interface for your IP tunnel.

Note
Cisco NX-OS supports a maximum of 8 IP tunnels.

Note
Use the no interface tunnel command to remove the tunnel interface and all associated configuration.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>no interface tunnel number</td>
<td>Deletes the tunnel interface and the associated configuration.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# no interface tunnel 1</td>
<td></td>
</tr>
<tr>
<td>description string</td>
<td>Configures a description for the tunnel.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)# description GRE tunnel</td>
<td></td>
</tr>
<tr>
<td>mtu value</td>
<td>Sets the MTU of IP packets sent on an interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)# mtu 1400</td>
<td></td>
</tr>
<tr>
<td>tunnel ttl value</td>
<td>Sets the tunnel time-to-live value. The range is from 1 to 255.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)# tunnel ttl 100</td>
<td></td>
</tr>
</tbody>
</table>

Note
Configuring a GRE tunnel or an IP-in-IP tunnel that uses a tunnel interface VRF that is different than the use-vrf for the tunnel destination is not supported. You need to use the same VRF for a tunnel interface and the tunnel destination.

Before you begin
You can configure the tunnel source and the tunnel destination in different VRFs. Ensure that you have enabled the tunneling feature.

SUMMARY STEPS

1. configure terminal
2. interface tunnel number
3. tunnel mode {gre ip | ipip {ip | decapsulate-any}}
4. tunnel source {ip-address | interface-name}
5. **tunnel destination** \{ip-address | host-name\}
6. **tunnel use-vrf** vrf-name
7. **show interfaces tunnel** number
8. **copy running-config startup-config**

## Detailed Steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><strong>configure terminal</strong></td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
| **Example:** | switche# configure terminal  
| | switche(config)# | |
| Step 2 | **interface tunnel** number | Creates a new tunnel interface. |
| **Example:** | switche(config)# interface tunnel 1  
| | switche(config-if)# | |
| Step 3 | **tunnel mode** \{gre ip | ipip ip | decapsulate-any\} | Sets this tunnel mode to GRE, ipip, or ipip decapsulate-only. The gre and ip keywords specify that GRE encapsulation over IP will be used. The ipip keyword specifies that IP-in-IP encapsulation will be used. The optional decapsulate-any keyword terminates IP-in-IP tunnels at one tunnel interface. This keyword creates a tunnel that will not carry any outbound traffic. However, remote tunnel endpoints can use a tunnel configured as their destination. |
| **Example:** | switche(config-if)# tunnel source  
| | ethernet 1/2 | |
| Step 4 | **tunnel source** \{ip-address | interface-name\} | Configures the source address for this IP tunnel. The source can be specified by IP address or logical interface name. |
| **Example:** | switche(config-if)# tunnel source  
| | ethernet 1/2 | |
| Step 5 | **tunnel destination** \{ip-address | host-name\} | Configures the destination address for this IP tunnel. The destination can be specified by IP address or logical host name. |
| **Example:** | switche(config-if)# tunnel destination  
| | 192.0.2.1 | |
| Step 6 | **tunnel use-vrf** vrf-name | (Optional) Uses the configured VRF to look up the tunnel IP destination address. |
| **Example:** | switche(config-if)# tunnel use-vrf blue | |
| Step 7 | **show interfaces tunnel** number | (Optional) Displays the tunnel interface statistics. |
| **Example:** | switche# show interfaces tunnel 1 | |
Creating an IP-in-IP Tunnel with a Netmask

Creating an IP-in-IP tunnel with a netmask allows you to specify a tunnel source subnet and a tunnel destination subnet, and decap the packet if it matches.

- The IP-in-IP decap-any tunnel receives encapsulated packets from any number of IP-in-IP tunnels.
- With the netmask feature, the switch receives packets from IP addresses which comply with the netmasks.

Notes for the netmask feature:

- Routing protocols are not supported on an IP-in-IP tunnel created with a netmask.
- Encap is not supported with the netmask feature; only decap from a set of sources in the same subnet is supported.

**SUMMARY STEPS**

1. `configure terminal`
2. `interface tunnel number`
3. `tunnel mode ipip [ip]`
4. `tunnel source ip-address / mask_length`
5. `tunnel destination ip-address / mask_length`
6. (Optional) `no shut`
7. `ip address ip-prefix/length`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 <code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

Example:

```
switch# configure terminal
switch(config)# interface tunnel 1
switch(config-if)# tunnel source ethenet 1/2
switch(config-if)# tunnel destination 192.0.2.1
switch(config-if)# copy running-config startup-config
```
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 2** | interface tunnel *number*  
*Example:*  
switch(config)# interface tunnel 5  
switch(config-if)# |
| **Purpose** | Creates a new tunnel interface. |
| **Step 3** | tunnel mode ipip [ip]  
*Example:*  
switch(config-if)# tunnel mode ipip  
switch(config-if)# |
| **Purpose** | Sets this tunnel mode to ipip.  
The *ipip* keyword specifies that IP-in-IP encapsulation will be used. |
| **Step 4** | tunnel source *ip-address / mask_length*  
*Example:*  
switch(config-if)# tunnel source 33.1.1.1 255.255.255.0  
switch(config-if)# |
| **Purpose** | Configures the source address for this IP tunnel. The source is specified by IP address and length of mask. |
| **Step 5** | tunnel destination *ip-address / mask_length*  
*Example:*  
switch(config-if)# tunnel destination 33.1.1.2 255.255.255.0  
switch(config-if)# |
| **Purpose** | Configures the destination address for this IP tunnel. The destination is specified by IP address and length of mask. |
| **Step 6** | (Optional) no shut  
*Example:*  
switch(config-if)# no shut  
switch(config-if)# |
| **Purpose** | Clears the interface. |
| **Step 7** | ip address *ip-prefix/length*  
*Example:*  
switch(config-if)# ip address 50.1.1.1/24  
switch(config-if)# |
| **Purpose** | Configures an IP address for this interface. |

**Example**

The following example shows how to create an IP-in-IP tunnel with a netmask.

```
switch(config)# interface tunnel 10  
switch(config-if)# tunnel mode ipip  
switch(config-if)# tunnel source 33.1.1.2/24  
switch(config-if)# tunnel destination 33.1.1.1/24  
switch(config-if)# no shut  
switch(config-if)# ip address 10.10.10.10/24  
switch(config-if)# end  
switch# show interface tunnel 10  
Tunnel10 is up  
Admin State: up  
Internet address is 10.10.10.10/24  
MTU 1476 bytes, BW 9 Kbit  
Tunnel protocol/transport IPIP/IP  
Tunnel source 33.1.1.2, destination 33.1.1.1  
Transport protocol is in VRF "default"  
Last clearing of "show interface" counters never  
Tx 0 packets output, 0 bytes  
Rx 0 packets input, 0 bytes
```
Configuring a Tunnel Interface

You can set a tunnel interface to GRE tunnel mode, ipip mode, or ipip decapsulate-only mode. GRE mode is the default tunnel mode.

Beginning with Cisco NX-OS Release 7.0(3)I6(1), the tunnel source direct and tunnel mode ipv6ipv6 decapsulate-any CLI commands are supported on Cisco Nexus 9000 Series switches.

The tunnel source direct and tunnel mode ipv6ipv6 decapsulate-any CLI commands are supported on Cisco Nexus 9000 Series switches.

---

Note

The tunnel source direct and tunnel mode ipv6ipv6 decapsulate-any CLI commands are not supported on Cisco Nexus 9500 platform switches with the Network Forwarding Engine (NFE).

The new CLI tunnel mode ipv6ipv6 decapsulate-any command is introduced to support IPv6 payload over IPv6 transport (IPv6inIPv6 packets). You can configure IP-in-IP tunnel decapsulation on directly connected IP addresses (for example, physical interface, port-channel, loopback, and SVI) using the new tunnel source direct CLI command.

**Before you begin**

Ensure that you have enabled the tunneling feature.

**SUMMARY STEPS**

1. configure terminal
2. interface tunnel number
3. tunnel mode {gre ip | ipip | {ip | decapsulate-any}}
4. (Optional) tunnel mode ipv6ipv6 decapsulate-any
5. tunnel source direct
6. show interfaces tunnel number
7. mtu value
8. copy running-config startup-config
### Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: switch# configure terminal switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface tunnel <em>number</em></td>
<td>Creates a new tunnel interface.</td>
</tr>
<tr>
<td>Example: switch(config)# interface tunnel 1 switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> tunnel mode {gre ip</td>
<td>ipip</td>
</tr>
<tr>
<td><strong>Step 4</strong> (Optional) tunnel mode ipv6ipv6 decapsulate-any</td>
<td>Supports IPv6 payload over IPv6 transport (IPv6inIPv6 packets). This step is applicable for IPv6 networks only. <strong>Note</strong> This command is not supported on Cisco Nexus 9500 platform switches.</td>
</tr>
<tr>
<td><strong>Step 5</strong> tunnel source direct</td>
<td>Configures IP-in-IP tunnel decapsulation on any directly connected IP addresses. This option is now supported only when the IP-in-IP decapsulation is used to source route the packets through the network. <strong>Note</strong> This command is not supported on Cisco Nexus 9500 platform switches with the Network Forwarding Engine (NFE).</td>
</tr>
<tr>
<td><strong>Step 6</strong> show interfaces tunnel <em>number</em></td>
<td>(Optional) Displays the tunnel interface statistics.</td>
</tr>
<tr>
<td>Example: switch(config-if)# show interfaces tunnel 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> mtu <em>value</em></td>
<td>Sets the maximum transmission unit (MTU) of IP packets sent on an interface. The range is from 64 to 9192 units.</td>
</tr>
</tbody>
</table>
### Purpose
When configuring tunnel mode ipip, the range is dependent on NX-OS release:
- 64 to 9192 units
- 64 to 9196 units

### Step 8
**copy running-config startup-config**

(Optional) Saves this configuration change.

---

### Example

This example shows how to create the tunnel interface to GRE:

```
switch# configure terminal
switch(config)# interface tunnel 1
switch(config-if)# tunnel mode gre ip
switch(config-if)# copy running-config startup-config
```

This example shows how to create an ipip tunnel:

```
switch# configure terminal
switch(config)# interface tunnel 1
switch(config-if)# tunnel mode ipip
switch(config-if)# mtu 1400
switch(config-if)# copy running-config startup-config
```

This example shows how to configure IP-in-IP tunnel decapsulation on directly connected IP addresses:

```
switch# configure terminal
switch(config)# interface tunnel 0
switch(config-if)# tunnel mode ipip ip
switch(config-if)# tunnel source direct
switch(config-if)# description IPinIP Decapsulation Interface
switch(config-if)# no shut
```

This example shows how to configure IP-in-IP tunnel decapsulation on IPv6 enabled networks:

```
! interface tunnel 1
   ipv6 address use-link-local-only <<< enable IPv6
tunnel mode ipv6ip6 decapsulate-any
tunnel source direct
description IPinIP Decapsulation Interface
mtu 1476
no shutdown
```

show running-config interface tunnel 1
```
interface Tunnel1
tunnel mode ipv6ip6 decapsulate-any
tunnel source direct
no shutdown
```

show interface tunnel 1
Configuring a GRE Tunnel

You can set a tunnel interface to GRE tunnel mode.

Note

Cisco NX-OS supports only the GRE protocol for IPV4 over IPV4.

Before you begin

Ensure that you have enabled the tunneling feature.

SUMMARY STEPS

1. configure terminal
2. interface tunnel number
3. tunnel mode gre ip
4. show interfaces tunnel number
5. copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td>Step 2 interface tunnel number</td>
<td>Creates a new tunnel interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# interface tunnel 1</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)#</td>
<td></td>
</tr>
<tr>
<td>Step 3 tunnel mode gre ip</td>
<td>Sets this tunnel mode to GRE.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)# tunnel mode gre ip</td>
<td></td>
</tr>
<tr>
<td>Step 4 show interfaces tunnel number</td>
<td>(Optional) Displays the tunnel interface statistics.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
Enabling Path MTU Discovery

Use the `tunnel path-mtu discovery` command to enable path MTU discovery on a tunnel.

**SUMMARY STEPS**

1. `tunnel path-mtu-discovery age-timer min`
2. `tunnel path-mtu-discovery min-mtu bytes`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>tunnel path-mtu-discovery age-timer min</code></td>
<td>Enables Path MTU Discovery (PMTUD) on a tunnel interface.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>switch(config-if)# tunnel path-mtu-discovery age-timer 25</code></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>tunnel path-mtu-discovery min-mtu bytes</code></td>
<td>Enables Path MTU Discovery (PMTUD) on a tunnel interface.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>switch(config-if)# tunnel path-mtu-discovery min-mtu 1500</code></td>
<td></td>
</tr>
</tbody>
</table>

Assigning VRF Membership to a Tunnel Interface

You can add a tunnel interface to a VRF.

**Before you begin**

Ensure that you have enabled the tunneling feature.

Assign the IP address for a tunnel interface after you have configured the interface for a VRF.

**SUMMARY STEPS**

1. `configure terminal`
2. `interface tunnel number`
3. `vrf member vrf-name`
4. `ip address ip-prefix/length`
5. show vrf [vrf-name] interface interface-type number  
6. copy running-config startup-config

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
</tbody>
</table>
| Example:          | switch# configure terminal  
                     | switch(config)# |
| **Step 2**        | interface tunnel number | Enters interface configuration mode. |
| Example:          | switch(config)# interface tunnel 0  
                     | switch(config-if)# |
| **Step 3**        | vrf member vrf-name | Adds this interface to a VRF. |
| Example:          | switch(config-if)# vrf member RemoteOfficeVRF |
| **Step 4**        | ip address ip-prefix/length | Configures an IP address for this interface. You must do this step after you assign this interface to a VRF. |
| Example:          | switch(config-if)# ip address 192.0.2.1/16 |
| **Step 5**        | show vrf [vrf-name] interface interface-type number | (Optional) Displays VRF information. |
| Example:          | switch(config-vrf)# show vrf Enterprise  
                     | interface tunnel 0 |
| **Step 6**        | copy running-config startup-config | (Optional) Saves this configuration change. |
| Example:          | switch# copy running-config startup-config |

**Example**  
This example shows how to add a tunnel interface to the VRF:  
switch# configure terminal  
switch(config)# interface tunnel 0  
switch(config-if)# vrf member RemoteOfficeVRF  
switch(config-if)# ip address 209.0.2.1/16  
switch(config-if)# copy running-config startup-config

### Verifying the IP Tunnel Configuration

To verify the IP tunnel configuration information, perform one of the following tasks:
### Command and Purpose

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show interface tunnel number</code></td>
<td>Displays the configuration for the tunnel interface (MTU, protocol, transport, and VRF). Displays input and output packets, bytes, and packet rates.</td>
</tr>
<tr>
<td><code>show interface tunnel number brief</code></td>
<td>Displays the operational status, IP address, encapsulation type, and MTU of the tunnel interface.</td>
</tr>
<tr>
<td><code>show interface tunnel number counters</code></td>
<td>Displays interface counters of input/output packets. Note: The byte count displayed with the interface counters include the internal header size.</td>
</tr>
<tr>
<td><code>show interface tunnel number description</code></td>
<td>Displays the configured description of the tunnel interface.</td>
</tr>
<tr>
<td><code>show interface tunnel number status</code></td>
<td>Displays the operational status of the tunnel interface.</td>
</tr>
<tr>
<td><code>show interface tunnel number status err-disabled</code></td>
<td>Displays the error disabled status of the tunnel interface.</td>
</tr>
</tbody>
</table>

### Configuration Examples for IP Tunneling

The following example shows a simple GRE tunnel. Ethernet 1/2 is the tunnel source for router A and the tunnel destination for router B. Ethernet interface 2/1 is the tunnel source for router B and the tunnel destination for router A.

**Router A:**

```bash
feature tunnel
interface tunnel 0
ip address 209.165.20.2/8
tunnel source ethernet 1/2
tunnel destination 192.0.2.2
tunnel mode gre ip
tunnel path-mtu-discovery 25 1500

interface ethernet 1/2
ip address 192.0.2.55/8
```

**Router B:**

```bash
feature tunnel
interface tunnel 0
ip address 209.165.20.1/8
tunnel source ethernet 2/1
tunnel destination 192.0.2.55
tunnel mode gre ip

interface ethernet 2/1
ip address 192.0.2.2/8
```
## Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Tunnel commands</td>
<td><em>Cisco Nexus 9000 Series NX-OS Interfaces Command Reference</em></td>
</tr>
</tbody>
</table>
CHAPTER 10

Configuring Q-in-Q VLAN Tunnels

- Information About Q-in-Q Tunnels, on page 315
- Licensing Requirements for Interfaces, on page 320
- Guidelines and Limitations, on page 320
- Guidelines and Limitations for Selective Q-in-Q with Multiple Provider VLANs, on page 322
- Configuring Q-in-Q Tunnels and Layer 2 Protocol Tunneling, on page 322
- Configuring Q-in-Q Double Tagging, on page 333
- Verifying the Q-in-Q Configuration, on page 335
- Configuration Examples for Q-in-Q and Layer 2 Protocol Tunneling, on page 335

Information About Q-in-Q Tunnels

This chapter describes how to configure IEEE 802.1Q-in-Q VLAN tunnels and Layer 2 protocol tunneling on Cisco NX-OS devices.

A Q-in-Q VLAN tunnel enables a service provider to segregate the traffic of different customers in their infrastructure, while still giving the customer a full range of VLANs for their internal use by adding a second 802.1Q tag to an already tagged frame.

Q-in-Q Tunneling

Business customers of service providers often have specific requirements for VLAN IDs and the number of VLANs to be supported. The VLAN ranges required by different customers in the same service-provider network might overlap, and the traffic of customers through the infrastructure might be mixed. Assigning a unique range of VLAN IDs to each customer would restrict customer configurations and could easily exceed the VLAN limit of 4096 of the 802.1Q specification.

Note

Q-in-Q is supported on port channels. To configure a port channel as an asymmetrical link, all ports in the port channel must have the same tunneling configuration.

Using the 802.1Q tunneling feature, service providers can use a single VLAN to support customers who have multiple VLANs. Customer VLAN IDs are preserved and the traffic from different customers is segregated within the service-provider infrastructure even when they appear to be on the same VLAN. The 802.1Q tunneling expands the VLAN space by using a VLAN-in-VLAN hierarchy and tagging the tagged packets.
A port configured to support 802.1Q tunneling is called a tunnel port. When you configure tunneling, you assign a tunnel port to a VLAN that is dedicated to tunneling. Each customer requires a separate VLAN, but that VLAN supports all of the customer’s VLANs.

Customer traffic that is tagged in the normal way with appropriate VLAN IDs come from an 802.1Q trunk port on the customer device and into a tunnel port on the service-provider edge switch. The link between the customer device and the edge switch is an asymmetric link because one end is configured as an 802.1Q trunk port and the other end is configured as a tunnel port. You assign the tunnel port interface to an access VLAN ID that is unique to each customer. See the figure below.

*Figure 33: 802.1Q-in-Q Tunnel Ports*

![Diagram of 802.1Q-in-Q Tunnel Ports](image)

Packets that enter the tunnel port on the service-provider edge switch, which are already 802.1Q-tagged with the appropriate VLAN IDs, are encapsulated with another layer of an 802.1Q tag that contains a VLAN ID that is unique to the customer. The original 802.1Q tag from the customer is preserved in the encapsulated packet. Therefore, packets that enter the service-provider infrastructure are double-tagged.

The outer tag contains the customer’s access VLAN ID (as assigned by the service provider), and the inner VLAN ID is the VLAN of the incoming traffic (as assigned by the customer). This double tagging is called tag stacking, Double-Q, or Q-in-Q as shown in the figure below.
By using this method, the VLAN ID space of the outer tag is independent of the VLAN ID space of the inner tag. A single outer VLAN ID can represent the entire VLAN ID space for an individual customer. This technique allows the customer’s Layer 2 network to extend across the service provider network, potentially creating a virtual LAN infrastructure over multiple sites.

**Note**
Hierarchical tagging, or multi-level dot1q tagging Q-in-Q, is not supported.

**Native VLAN Hazard**

When configuring 802.1Q tunneling on an edge switch, you must use 802.1Q trunk ports for sending out packets into the service-provider network. However, packets that go through the core of the service-provider network might be carried through 802.1Q trunks, ISL trunks, or nontrunking links. When 802.1Q trunks are used in these core switches, the native VLANs of the 802.1Q trunks must not match any native VLAN of the dot1q-tunnel port on the same switch because traffic on the native VLAN is not tagged on the 802.1Q transmitting trunk port.

In the figure below, VLAN 40 is configured as the native VLAN for the 802.1Q trunk port from Customer X at the ingress edge switch in the service-provider network (Switch B). Switch A of Customer X sends a tagged packet on VLAN 30 to the ingress tunnel port of Switch B in the service-provider network that belongs to access VLAN 40. Because the access VLAN of the tunnel port (VLAN 40) is the same as the native VLAN of the edge-switch trunk port (VLAN 40), the 802.1Q tag is not added to tagged packets that are received from the tunnel port. The packet carries only the VLAN 30 tag through the service-provider network to the trunk port of the egress-edge switch (Switch C) and is misdirected through the egress switch tunnel port to Customer Y.
These are a couple ways to solve the native VLAN problem:

- Configure the edge switch so that all packets going out an 802.1Q trunk, including the native VLAN, are tagged by using the `vlan dot1q tag native` command. If the switch is configured to tag native VLAN packets on all 802.1Q trunks, the switch accepts untagged packets but sends only tagged packets.

  **Note** The `vlan dot1q tag native` command is a global command that affects the tagging behavior on all trunk ports.

- Ensure that the native VLAN ID on the edge switch trunk port is not within the customer VLAN range. For example, if the trunk port carries traffic of VLANs 100 to 200, assign the native VLAN a number outside that range.

### Information About Layer 2 Protocol Tunneling

Customers at different sites connected across a service-provider network need to run various Layer 2 protocols to scale their topology to include all remote sites, as well as the local sites. The Spanning Tree Protocol (STP) must run properly, and every VLAN should build a proper spanning tree that includes the local site and all remote sites across the service-provider infrastructure. The Cisco Discovery Protocol (CDP) must be able to discover neighboring Cisco devices from local and remote sites, and the VLAN Trunking Protocol (VTP) must provide consistent VLAN configuration throughout all sites in the customer network.

You can configure the switch to allow multi-tagged BPDUs on a tunnel port. If you enable the `l2protocol tunnel allow-double-tag` command, when a multi-tagged customer BPDU enters the tunnel port, the original 802.1Q tags from the customer traffic is preserved and an outer VLAN tag (customer’s access VLAN ID, as assigned by the service-provider) is added in the encapsulated packet. Therefore, BPDU packets that enter the service-provider infrastructure are multi tagged. When the BPDUs leave the service-provider network, the outer tag is removed and the original multi-tagged BPDU is sent to the customer network.
When protocol tunneling is enabled, edge switches on the inbound side of the service-provider infrastructure encapsulate Layer 2 protocol packets with a special MAC address and send them across the service-provider network. Core switches in the network do not process these packets, but forward them as normal packets. Bridge protocol data units (BPDUs) for CDP, STP, or VTP cross the service-provider infrastructure and are delivered to customer switches on the outbound side of the service-provider network. Identical packets are received by all customer ports on the same VLANs.

If protocol tunneling is not enabled on 802.1Q tunneling ports, remote switches at the receiving end of the service-provider network do not receive the BPDUs and cannot properly run STP, CDP, 802.1X, and VTP. When protocol tunneling is enabled, Layer 2 protocols within each customer’s network are totally separate from those running within the service-provider network. Customer switches on different sites that send traffic through the service-provider network with 802.1Q tunneling achieve complete knowledge of the customer’s VLAN.

**Note**

Layer 2 protocol tunneling works by tunneling BPDUs in the software. A large number of BPDUs that come into the supervisor will cause the CPU load to go up. You might need to make use of software rate limiters to reduce the load on the supervisor CPU. See Configuring Thresholds for Layer 2 Protocol Tunnel Ports, on page 332.

For example, in the figure below, Customer X has four switches in the same VLAN that are connected through the service-provider network. If the network does not tunnel BPDUs, switches on the far ends of the network cannot properly run the STP, CDP, 802.1X, and VTP protocols.

*Figure 36: Layer 2 Protocol Tunneling*

In the preceding example, STP for a VLAN on a switch in Customer X, Site 1 will build a spanning tree on the switches at that site without considering convergence parameters based on Customer X’s switch in Site 2.

The figure below shows the resulting topology on the customer’s network when BPU tunneling is not enabled.
Selective Q-in-Q with Multiple Provider VLANs

Selective Q-in-Q with multiple provider VLANs is a tunneling feature that allows user-specific range of customer VLANs on a port to be associated with one specific provider VLAN and enables you to have multiple customer VLAN to provider VLAN mappings on a port. Packets that come in with a VLAN tag that matches any of the configured customer VLANs on the port are tunneled across the fabric using the properties of the service provider VLAN. The encapsulated packet carries the customer VLAN tag as part of the Layer 2 header of the inner packet.

Licensing Requirements for Interfaces

Interfaces do not require a license.

Guidelines and Limitations

Q-in-Q tunnels and Layer 2 tunneling have the following configuration guidelines and limitations:

- Mapping multiple VLANs is supported.
- Multi-tagged BPDUs are supported on the Cisco Nexus 93108TC-EX and 93180YC-EX switches. We support up to three tags.
- Selective Q-in-Q tunneling is not supported with multi-tagged BPDU.
- Only multi-tagged CDP and STP BPDUs are supported.
- The inner-most tag must always be 0x8100.
- Multiple selective Q-in-Q tags are not supported. That is, Q-in-Q does not support multiple SP tags on a single interface.
- show commands with the internal keyword are not supported.
- Switches in the service-provider network must be configured to handle the increase in MTU size due to Q-in-Q tagging.
• MAC address learning for Q-in-Q tagged packets is based on the outer VLAN (Service Provider VLAN) tag. Packet forwarding issues might occur in deployments where a single MAC address is used across multiple inner (customer) VLANs.

• Layer 3 and higher parameters cannot be identified in tunnel traffic (for example, Layer 3 destination and source addresses). Tunneled traffic cannot be routed.

• Cisco Nexus 9000 Series devices can provide only MAC-layer ACL/QoS for tunnel traffic (VLAN IDs and src/dest MAC addresses).

• You should use MAC address-based frame distribution.

• Asymmetrical links do not support the Dynamic Trunking Protocol (DTP) because only one port on the link is a trunk. You must configure the 802.1Q trunk port on an asymmetrical link to trunk unconditionally.

• You cannot configure the 802.1Q tunneling feature on ports that are configured to support private VLANs. Private VLAN are not required in these deployments.

• You must disable IGMP snooping on the tunnel VLANs.

• Control Plane Policing (CoPP) is not supported.

• You should enter the vlan dot1Q tag native command to maintain the tagging on the native VLAN and drop untagged traffic. This command prevents native VLAN misconfigurations.

• You must manually configure the 802.1Q interfaces to be edge ports.

• IGMP snooping is not supported on the inner VLAN.

• Q-in-Q is not supported on the uplink ports of Cisco Nexus 9332PQ, 9372PX, 9372TX, and 93120TX switches and Cisco Nexus 9396PX, 9396TX, and 93128TX switches with the N9K-M6PQ or N9K-M12PQ generic expansion module (GEM).

• Q-in-Q tunnels might be affected by the limitations of the Application Leaf Engine (ALE) uplink ports on Cisco Nexus 9300 and 9500 Series devices: Limitations for ALE Uplink Ports

• Q-in-Q tunneling is not supported on Application Spine Engine 2 (ASE2) and Application Spine Engine 3 (ASE3) based Cisco Nexus switches.

• Layer 2 protocol tunnelling with Link Aggregation Control Protocol (LACP) is not supported.

• Q-in-Q tagging is not supported.

• Layer 2 protocol tunneling is not supported on Cisco Nexus 9500 Series switches with N9K-X9636C-R, N9K-X9636Q-R, N9K-X9636C-RX line cards.

• Cisco Nexus 9500 Series switches with N9K-X9636C-R, N9K-X9636Q-R, N9K-X9636C-RX line cards, Q-in-Q is supported only on port or port-channel Layer 2 Access VLAN Edge devices.

• FEX is not supported for Q-in-Q

• If the command l2protocol tunne stp is configured on a tunnel interface, the VLAN that you configure on the service provider must be different from that of the customer network.
Guidelines and Limitations for Selective Q-in-Q with Multiple Provider VLANs

- For selective Q-in-Q with multiple provider VLANs, all the existing limitations and guidelines for selective Q-in-Q apply.
- Selective Q-in-Q with multiple provider VLANs feature is supported on Nexus 9300-EX, 9300-FX, and 9300-FX2 platforms.
- When you enable multiple provider VLANs on a vPC port channel, you must make sure that the configuration is consistent across the vPC peers.
- We recommend enabling “system dot1q tunnel transit” when running selective Q-in-Q with multiple provider VLAN features on a vPC setup.
- We recommended not to allow provider VLANs on a regular trunk.
- Only allow native VLANs and provider VLANs on the trunk interface allowed VLAN list of a multiple provider VLAN interface.
- Port to VLAN mappings (for example: switchport vlan mapping 10 20) is not supported on a port that is configured for selective Q-in-Q with multiple provider VLANs.
- Private VLAN is not supported on a port that is configured for selective Q-in-Q with multiple provider VLANs.
- Only Layer 2 switching is supported.
- Routing on provider VLANs is not supported.
- FEX is not supported for selective Q-in-Q with multiple provider VLANs.
- Selective Q-in-Q with multiple provider VLANs commands not DME-ized

Configuring Q-in-Q Tunnels and Layer 2 Protocol Tunneling

Creating a 802.1Q Tunnel Port

You create the dot1q-tunnel port using the switchport mode command.

**Note**

You must set the 802.1Q tunnel port to an edge port with the spanning-tree port type edge command. The VLAN membership of the port is changed using the switchport access vlan vlan-id command.

You should disable IGMP snooping on the access VLAN allocated for the dot1q-tunnel port to allow multicast packets to traverse the Q-in-Q tunnel.

The following CLI is mandatory only on LSE, EX, FX, FX2 based Cisco Nexus 9000 Series switches. For seamless packet forwarding and preservation of all VLAN tags on pure transit boxes in the SP cloud that have
no Q-in-Q encapsulation or decapsulation requirement, configure the CLI command `system dot1q-tunnel transit`. To remove the CLI, use `no system dot1q-tunnel transit` CLI command.

The caveats with the CLI that is executed on the switches are:

- L2 frames that egress out of the trunk ports are tagged even on the native VLAN on the port.
- Any other tunneling mechanisms, for instance, VXLAN and MPLS may not work with the CLI configured.

**Before you begin**

You must first configure the interface as a switchport.

**SUMMARY STEPS**

1. `switch# configure terminal`
2. `switch(config)# interface ethernet slot/port`
3. `switch(config-if)# switchport`
4. `switch(config-if)# switchport mode dot1q-tunnel`
5. (Optional) `switch(config-if)# no switchport mode dot1q-tunnel`
6. `switch(config-if)# exit`
7. (Optional) `switch(config)# show dot1q-tunnel [interface if-range]`
8. (Optional) `switch(config)# no shutdown`
9. (Optional) `switch(config)# copy running-config startup-config`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# interface ethernet slot/port</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config-if)# switchport</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>switch(config-if)# switchport mode dot1q-tunnel</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>(Optional) switch(config-if)# no switchport mode dot1q-tunnel</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>switch(config-if)# exit</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>(Optional) switch(config)# show dot1q-tunnel [interface if-range]</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>(Optional) switch(config)# no shutdown</td>
</tr>
</tbody>
</table>
### Example

This example shows how to create an 802.1Q tunnel port:

```plaintext
switch# configure terminal
switch(config)# interface ethernet 7/1
switch(config-if)# switchport
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# exit
switch(config)# exit

switch#
```

### Configuring VLAN Mapping for Selective Q-in-Q on a 802.1Q Tunnel Port

To configure VLAN mapping for selective Q-in-Q on a 802.1Q tunnel port, complete the following steps.

**Note**

You cannot configure one-to-one mapping and selective Q-in-Q on the same interface.

#### SUMMARY STEPS

1. `switch# configure terminal`
2. `switch(config)# interface interface-id`
3. `switch(config-if)# switchport mode dot1q-tunnel`
4. `switch(config-if)# switchport vlan mapping vlan-id-range dot1q-tunnel outer vlan-id`
5. `switch(config-if)# exit`
6. `switch# show interfaces interface-id vlan mapping`
7. `switch# copy running-config startup-config`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# interface interface-id</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Enters interface configuration mode for the interface connected to the service provider network. You can enter a physical interface or an EtherChannel port channel.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config-if)# switchport mode dot1q-tunnel</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Configure the interface as a tunnel port.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>switch(config-if)# switchport vlan mapping vlan-id-range dot1q-tunnel outer vlan-id</td>
</tr>
</tbody>
</table>
| **Purpose** | Enters the VLAN IDs to be mapped:  
  - vlan-id-range—The customer VLAN ID range (C-VLAN) entering the switch from the customer |
### Configuring Selective Q-in-Q with Multiple provider VLANs

**Before you begin**

You must configure provider VLANs.

**SUMMARY STEPS**

1. `switch# configure terminal`
2. `switch(config)# interface interface-id`
3. `switch(config-if)# switchport`
4. `switch(config-if)# switchport mode trunk`
5. `switch(config-if)# switchport trunk native vlan vlan-id`
6. `switch(config-if)# switchport vlan mapping vlan-id-range dot1q-tunnel outer vlan-id`
7. `switch(config-if)# switchport trunk allowed vlan vlan_list`
8. `switch(config-if)# exit`
9. `switch(config-if)# show interfaces interface-id vlan mapping`

Use the `no switchport vlan mapping vlan-id-range dot1q-tunnel outer vlan-id` command to remove the VLAN mapping configuration.

The following example shows how to configure selective Q-in-Q mapping on the port so that traffic with a C-VLAN ID of 1 to 5 enters the switch with an S-VLAN ID of 100. The traffic of any other VLAN IDs is dropped.

**Example**

```plaintext
switch(config)# interface gigabiethernet0/1
switch(config-if)# switchport vlan mapping 1-5 dot1q-tunnel 100
```

```
Switch(config-if)# exit
```
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# interface interface-id</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config-if)# switchport</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>switch(config-if)# switchport mode trunk</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>switch(config-if)# switchport trunk native vlan vlan-id</td>
</tr>
</tbody>
</table>
| **Step 6**        | switch(config-if)# switchport vlan mapping vlan-id-range dot1q-tunnel outer vlan-id | Enter the VLAN IDs to be mapped:  
  - **vlan-id-range**—The customer VLAN ID range (C-VLAN) entering the switch from the customer network. The range is from 1 to 4094. You can enter a string of VLAN-IDs.  
  - **outer vlan-id**—Enter the outer VLAN ID (S-VLAN) of the service provider network. The range is from 1 to 4094. |
| **Step 7**        | switch(config-if)# switchport trunk allowed vlan vlan_list | Sets the allowed VLANs for the trunk interface. |
| **Step 8**        | switch(config-if)# exit | Exits the configuration mode. |
| **Step 9**        | switch(config-if)# show interfaces interface-id vlan mapping | Verifies the mapping configuration. |

The following example shows how to configure selective Q-in-Q with multiple provider VLANs:

### Example

```
switch# sh run int e1/1

interface Ethernet1/1
    switchport
    switchport mode trunk
    switchport trunk native vlan 2
    switchport vlan mapping 3-400 dot1q-tunnel 400
    switchport vlan mapping 401-800 dot1q-tunnel 401
    switchport vlan mapping 801-1200 dot1q-tunnel 10
    switchport vlan mapping 1201-1600 dot1q-tunnel 1400
    switchport vlan mapping 1601-2000 dot1q-tunnel 9
    switchport vlan mapping 2001-2400 dot1q-tunnel 3000
    switchport vlan mapping 2401-2800 dot1q-tunnel 2099
    switchport vlan mapping 2801-3200 dot1q-tunnel 2800
    switchport vlan mapping 3201-3600 dot1q-tunnel 3967
    switchport vlan mapping 3601-4000 dot1q-tunnel 600
    switchport trunk allowed vlan 2,9-10,400-401,600,1400,2099,2800,3000,3967
```
Changing the EtherType for Q-in-Q

You can change the 802.1Q EtherType value to be used for Q-in-Q encapsulation.

---

**Note**

You must set the EtherType only on the egress trunk interface that carries double tagged frames (the trunk interface that connects the service providers). If you change the EtherType on one side of the trunk, you must set the same value on the other end of the trunk (symmetrical configuration).

---

**Caution**

The EtherType value you set affects all the tagged packets that go out on the interface (not just Q-in-Q packets).

---

**SUMMARY STEPS**

1. switch# configure terminal
2. switch(config)# interface ethernet slot/port
3. switch(config-if)# switchport
4. switch(config-if)# switchport dot1q ethertype value
5. (Optional) switch(config-if)# no switchport dot1q ethertype
6. switch(config-if)# exit
7. (Optional) switch(config)# no shutdown
8. (Optional) switch(config)# copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2 switch(config)# interface ethernet slot/port</td>
<td>Specifies an interface to configure, and enters interface configuration mode.</td>
</tr>
<tr>
<td>Step 3 switch(config-if)# switchport</td>
<td>Sets the interface as a Layer 2 switching port.</td>
</tr>
<tr>
<td>Step 4 switch(config-if)# switchport dot1q ethertype value</td>
<td>Sets the EtherType for the Q-in-Q tunnel on the port.</td>
</tr>
<tr>
<td>Step 5 (Optional) switch(config-if)# no switchport dot1q ethertype</td>
<td>Resets the EtherType on the port to the default value of 0x8100.</td>
</tr>
<tr>
<td>Step 6 switch(config-if)# exit</td>
<td>Exits configuration mode.</td>
</tr>
<tr>
<td>Step 7 (Optional) switch(config)# no shutdown</td>
<td>Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.</td>
</tr>
<tr>
<td>Step 8 (Optional) switch(config)# copy running-config startup-config</td>
<td>Copies the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>

Example

This example shows how to create an 802.1Q tunnel port:

```
switch# configure terminal
switch(config)# interface ethernet 7/1
switch(config-if)# switchport
switch(config-if)# switchport dot1q ethertype 0x9100
switch(config-if)# exit
switch(config)# exit
switch# show dot1q-tunnel
```

Configuring Custom EtherType or Tag Protocol Identifier (TPID)

The switch uses a default ethertype of 0x8100 for 802.1Q and Q-in-Q encapulations. You can configure EtherTypes 0x9100, 0x9200 and 0x88a8 on a per port basis by enabling the dot1q ethertype command on the switchport interface. You can configure a custom EtherType field value on a port to support network devices that do not use the standard 0x8100 EtherType field value on 802.1Q-tagged or 802.1p-tagged frames.
You must set the EtherType or TPID only on the egress trunk interface that carries double tagged frames. EtherType value impacts all the tagged packets that go out on the interface (on both Q-in-Q and 802.1Q packets).

**SUMMARY STEPS**

1. `switch# configure terminal`
2. `switch(config)# interface ethernet slot/port`
3. `switch(config-if)# switchport`
4. `switch(config-if)# switchport mode`
5. `switch(config-if)# switchport dot1q ethertype value`
6. (Optional) `switch(config-if)# switchport access vlan value`
7. `switch(config-if)# exit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# interface ethernet slot/port</td>
<td>Specifies an interface to configure, and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong> switch(config-if)# switchport</td>
<td>Sets the interface as a Layer 2 switching port.</td>
</tr>
<tr>
<td><strong>Step 4</strong> switch(config-if)# switchport mode</td>
<td>Sets the interface as a Layer 2 switching port mode.</td>
</tr>
<tr>
<td><strong>Step 5</strong> switch(config-if)# switchport dot1q ethertype value</td>
<td>Sets the EtherType for the Q-in-Q tunnel on the port.</td>
</tr>
<tr>
<td><strong>Step 6</strong> (Optional) switch(config-if)# switchport access vlan value</td>
<td>Sets the interface access VLAN.</td>
</tr>
<tr>
<td><strong>Step 7</strong> switch(config-if)# exit</td>
<td>Exits configuration mode.</td>
</tr>
</tbody>
</table>

**Example**

This example shows how to configure custom ethertype on an 802.1Q tunnel port:

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# switchport
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport dot1q ethertype 0x9100
switch(config-if)# switchport dot1q ethertype 0x9100
switch(config-if)# switchport access vlan 30
switch(config-if)# exit
switch(config)# exit
```
Enabling the Layer 2 Protocol Tunnel

You can enable protocol tunneling on the 802.1Q tunnel port.

### SUMMARY STEPS

1. `switch# configure terminal`
2. `switch(config)# interface ethernet slot/port`
3. `switch(config-if)# switchport`
4. `switch(config-if)# switchport mode dot1q-tunnel`
5. `switch(config-if)# l2protocol tunnel [cdp | stp | lacp | lldp | vtp]`
6. (Optional) `switch(config-if)# no l2protocol tunnel [cdp | stp | lacp | lldp | vtp]`
7. `switch(config-if)# exit`
8. (Optional) `switch(config)# no shutdown`
9. (Optional) `switch(config)# copy running-config startup-config`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>switch# configure terminal</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>switch(config)# interface ethernet slot/port</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>switch(config-if)# switchport</code></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><code>switch(config-if)# switchport mode dot1q-tunnel</code></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>`switch(config-if)# l2protocol tunnel [cdp</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>(Optional) `switch(config-if)# no l2protocol tunnel [cdp</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><code>switch(config-if)# exit</code></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>(Optional) <code>switch(config)# no shutdown</code></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>(Optional) <code>switch(config)# copy running-config startup-config</code></td>
</tr>
</tbody>
</table>

**Note**

Layer 2 protocol tunnel for LACP and LLDP is only supported on Cisco Nexus N9K-X9732C-EXM modules.
Example

This example shows how to enable protocol tunneling on an 802.1Q tunnel port:

```
switch# configure terminal
switch(config)# interface ethernet 7/1
switch(config-if)# switchport
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# l2protocol tunnel stp
switch(config-if)# exit
switch(config)# exit
```

Configuring Global CoS for L2 Protocol Tunnel Ports

You can specify a Class of Service (CoS) value globally so that ingress BPDUs on the tunnel ports are encapsulated with the specified class.

**SUMMARY STEPS**

1.  switch# configure terminal
2.  switch(config)# l2protocol tunnel cos value
3.  (Optional) switch(config)# no l2protocol tunnel cos
4.  switch(config)# exit
5.  (Optional) switch# no shutdown
6.  (Optional) switch# copy running-config startup-config

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Specifies a global CoS value on all Layer 2 protocol tunneling ports. The default cos-value is 5.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Sets the global CoS value to default.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Exits configuration mode.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.</td>
</tr>
<tr>
<td>Step 6</td>
<td>Copies the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>
Example
This example shows how to specify a global CoS value for the purpose of Layer 2 protocol tunneling:

```
switch# configure terminal
switch(config)# l2protocol tunnel cos 6
switch(config)# exit
```

Configuring Thresholds for Layer 2 Protocol Tunnel Ports

You can specify the port drop and shutdown value for a Layer 2 protocol tunneling port.

**SUMMARY STEPS**

1. `switch# configure terminal`
2. `switch(config)# interface ethernet slot/port`
3. `switch(config-if)# switchport`
4. `switch(config-if)# switchport mode dot1q-tunnel`
5. `switch(config-if)# l2protocol tunnel drop-threshold [cdp | stp | vtp] packets-per-sec`
6. (Optional) `switch(config-if)# no l2protocol tunnel drop-threshold [cdp | stp | vtp]`
7. `switch(config-if)# l2protocol tunnel shutdown-threshold [cdp | stp | vtp] packets-per-sec`
8. (Optional) `switch(config-if)# no l2protocol tunnel shutdown-threshold [cdp | stp | vtp]`
9. `switch(config-if)# exit`
10. (Optional) `switch(config)# no shutdown`
11. (Optional) `switch(config)# copy running-config startup-config`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# interface ethernet slot/port</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config-if)# switchport</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>switch(config-if)# switchport mode dot1q-tunnel</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>switch(config-if)# l2protocol tunnel drop-threshold [cdp</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>(Optional) switch(config-if)# no l2protocol tunnel drop-threshold [cdp</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>switch(config-if)# l2protocol tunnel shutdown-threshold [cdp</td>
</tr>
</tbody>
</table>
### Configuring Q-in-Q Double Tagging

Enable multi-tagging for STP and CDP BPDU.

#### SUMMARY STEPS

1. `configure terminal`
2. `interface interface`
3. `switchport`
4. `switchport mode dot1q-tunnel`
5. `l2protocol tunnel [cdp | stp]`
6. (Optional) `no l2protocol tunnel [cdp | stp]`
7. `l2protocol tunnel allow-double-tag`
8. (Optional) `no l2protocol tunnel allow-double-tag`
9. `exit`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure terminal</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>switch# configure terminal</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>interface interface</code>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>switch(config)# interface ethernet 7/1</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>switchport</code>&lt;br&gt;<strong>Example:</strong></td>
</tr>
</tbody>
</table>

---

Purpose Command or Action | Purpose
---|---
 | you can specify CDP, STP, or VTP. Valid values for the packets is from 1 to 4096. 
**Step 8** | (Optional) `switch(config-if)# no l2protocol tunnel shutdown-threshold [cdp | stp | vtp]` | Resets the threshold values to 0 and disables the shutdown threshold. |
**Step 9** | `switch(config-if)# exit` | Exits configuration mode. |
**Step 10** | (Optional) `switch(config)# no shutdown` | Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state. |
**Step 11** | (Optional) `switch(config)# copy running-config startup-config` | Copies the running configuration to the startup configuration. |
### Configuring Q-in-Q VLAN Tunnels

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch(config-if)# switchport</td>
<td>Creates an 802.1Q tunnel on the port. The port goes down and reinitializes (port flap) when the interface mode is changed. BPDU filtering is enabled and CDP is disabled on tunnel interfaces.</td>
</tr>
<tr>
<td><strong>Step 4</strong> switchport mode dot1q-tunnel</td>
<td>Enables Layer 2 protocol tunneling. Optionally, you can enable CDP or STP.</td>
</tr>
<tr>
<td>Example: switch(config-if)# switchport mode dot1q-tunnel</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> l2protocol tunnel [cdp</td>
<td>stp]</td>
</tr>
<tr>
<td>Example: switch(config-if)# l2protocol tunnel cdp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> (Optional) no l2protocol tunnel [cdp</td>
<td>stp]</td>
</tr>
<tr>
<td>Example: switch(config-if)# no l2protocol tunnel stp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> l2protocol tunnel allow-double-tag</td>
<td>Enables multi-tagging for STP and CDP BPDUs on the interface.</td>
</tr>
<tr>
<td>Example: switch(config-if)# l2protocol tunnel allow-double-tag</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> (Optional) no l2protocol tunnel allow-double-tag</td>
<td>Disables protocol tunneling.</td>
</tr>
<tr>
<td>Example: switch(config-if)# no l2protocol tunnel allow-double-tag</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> exit</td>
<td>Exits configuration mode.</td>
</tr>
<tr>
<td>Example: switch(config-if)# exit</td>
<td></td>
</tr>
</tbody>
</table>

### Example

This example shows how to enable multi-tagging for STP and CDP BPDUs:

```
switch# configure terminal
switch(config)# interface ethernet 7/1
switch(config-if)# switchport
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# l2protocol tunnel cdp
switch(config-if)# l2protocol tunnel stp
switch(config-if)# l2protocol tunnel allow-double-tag
switch(config-if)# exit
switch(config)# exit
switch#
```
Verifying the Q-in-Q Configuration

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>clear l2protocol tunnel counters [interface if-range]</code></td>
<td>Clears all the statistics counters. If no interfaces are specified, the Layer 2 protocol tunnel statistics are cleared for all interfaces.</td>
</tr>
<tr>
<td><code>show dot1q-tunnel [interface if-range]</code></td>
<td>Displays a range of interfaces or all interfaces that are in dot1q-tunnel mode.</td>
</tr>
<tr>
<td>`show l2protocol tunnel [interface if-range</td>
<td>vlan vlan-id]`</td>
</tr>
<tr>
<td><code>show l2protocol tunnel summary</code></td>
<td>Displays a summary of all ports that have Layer 2 protocol tunnel configurations.</td>
</tr>
<tr>
<td><code>show running-config l2pt</code></td>
<td>Displays the current Layer 2 protocol tunnel running configuration.</td>
</tr>
</tbody>
</table>

Configuration Examples for Q-in-Q and Layer 2 Protocol Tunneling

This example shows a service provider switch that is configured to process Q-in-Q for traffic coming in on Ethernet 7/1. A Layer 2 protocol tunnel is enabled for STP BPDUs. The customer is allocated VLAN 10 (outer VLAN tag).

```bash
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# vlan 10
switch(config-vlan)# no shutdown
switch(config-vlan)# no ip igmp snooping
switch(config-vlan)# exit
switch(config)# interface ethernet 7/1
switch(config-if)# switchport
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport access vlan 10
switch(config-if)# spanning-tree port type edge
switch(config-if)# l2protocol tunnel stp
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)# exit
switch#
```
Network Address Translation Overview

Network Address Translation (NAT) enables private IP internetworks that use nonregistered IP addresses to connect to the Internet. NAT operates on a device, usually connecting two networks, and translates private (not globally unique) IP addresses in the internal network into legal IP addresses before packets are forwarded to another network. You can configure NAT to advertise only one IP address for the entire network to the outside world. This ability provides additional security, effectively hiding the entire internal network behind one IP address.

A device configured with NAT has at least one interface to the inside network and one to the outside network. In a typical environment, NAT is configured at the exit router between a stub domain and a backbone. When a packet leaves the domain, NAT translates the locally significant source IP address into a globally unique IP address. When a packet enters the domain, NAT translates the globally unique destination IP address into a local IP address. If more than one exit point exists, NAT configured at each point must have the same translation table.

NAT is described in RFC 1631.
Information About Static NAT

Static Network Address Translation (NAT) allows the user to configure one-to-one translations of the inside local addresses to the outside global addresses. It allows both IP addresses and port number translations from the inside to the outside traffic and the outside to the inside traffic. The Cisco Nexus device supports Hitless NAT, which means that you can add or remove a NAT translation in the NAT configuration without affecting the existing NAT traffic flows.

Static NAT creates a fixed translation of private addresses to public addresses. Because static NAT assigns addresses on a one-to-one basis, you need an equal number of public addresses as private addresses. Because the public address is the same for each consecutive connection with static NAT, and a persistent translation rule exists, static NAT enables hosts on the destination network to initiate traffic to a translated host if an access list exists that allows it.

With dynamic NAT and Port Address Translation (PAT), each host uses a different address or port for each subsequent translation. The main difference between dynamic NAT and static NAT is that static NAT allows a remote host to initiate a connection to a translated host if an access list exists that allows it, while dynamic NAT does not.

The figure shows a typical static NAT scenario. The translation is always active so both translated and remote hosts can originate connections, and the mapped address is statically assigned by the static command.

These are key terms to help you understand static NAT:

- NAT inside interface—The Layer 3 interface that faces the private network.
- NAT outside interface—The Layer 3 interface that faces the public network.
- Local address—Any address that appears on the inside (private) portion of the network.
- Global address—Any address that appears on the outside (public) portion of the network.
- Legitimate IP address—An address that is assigned by the Network Information Center (NIC) or service provider.
• Inside local address—The IP address assigned to a host on the inside network. This address does not need to be a legitimate IP address.

• Outside local address—The IP address of an outside host as it appears to the inside network. It does not have to be a legitimate address, because it is allocated from an address space that can be routed on the inside network.

• Inside global address—A legitimate IP address that represents one or more inside local IP addresses to the outside world.

• Outside global address—The IP address that the host owner assigns to a host on the outside network. The address is a legitimate address that is allocated from an address or network space that can be routed.

Dynamic NAT Overview

Dynamic Network Address Translation (NAT) translates a group of real IP addresses into mapped IP addresses that are routable on a destination network. Dynamic NAT establishes a one-to-one mapping between unregistered and registered IP addresses; however, the mapping can vary depending on the registered IP address that is available at the time of communication.

A dynamic NAT configuration automatically creates a firewall between your internal network and outside networks or the Internet. Dynamic NAT allows only connections that originate inside the stub domain—a device on an external network cannot connect to devices in your network, unless your device has initiated the contact.

Dynamic NAT translations do not exist in the NAT translation table until a device receives traffic that requires translation. Dynamic translations are cleared or timed out when not in use to make space for new entries. Usually, NAT translation entries are cleared when the ternary content addressable memory (TCAM) entries are limited. The default minimum timeout for dynamic NAT translations is 30 minutes.

Note

The `ip nat translation sampling-timeout` command is not supported. Statistics are collected every 60 seconds for the installed NAT policies. These statistics are used to determine if the flow is active or not.

Dynamic NAT supports Port Address Translation (PAT) and access control lists (ACLs). PAT, also known as overloading, is a form of dynamic NAT that maps multiple unregistered IP addresses to a single registered IP address by using different ports. Your NAT configuration can have multiple dynamic NAT translations with same or different ACLs. However, for a given ACL, only one interface can be specified.

Timeout Mechanisms

The following NAT translation timeout timers are supported on the switch:

- `timeout`—Timeout value for dynamic NAT translations.
  The timeout value ranges from 60 seconds to 172800 seconds, including the sampling-timeout.

The `udp-timeout` and the `timeout` value timers are triggered after the timeout configured for the `ip nat translation sampling-timeout` command expires.
There are three different options that can be configured for aging:

- **Time-out**: This is applicable for all type of flows (both TCP and UDP).
- **TCP TIME-OUT**: This is applicable for only TCP flows.
- **UDP TIME-OUT**: This is applicable for only UDP flows.

Note

When you create dynamic entries without timeouts configured, they take the default timeout of one hour (60 minutes). If you enter the `clear ip nat translations all` command after configuring timeouts, the configured timeout take effect. A timeout can be configured from 60 to 172800 seconds.

**NAT Inside and Outside Addresses**

NAT inside refers to networks owned by an organization that must be translated. When NAT is configured, hosts within this network will have addresses in one space (known as the local address space) that will appear to those outside the network as being in another space (known as the global address space).

Similarly, NAT outside refers to those networks to which the stub network connects. They are not generally under the control of the organization. Hosts in outside networks can be subject to translation and can have local and global addresses.

NAT uses the following definitions:

- **Local address**—A local IP address that appears on the inside of a network.
- **Global address**—A global IP address that appears on the outside of a network.
- **Inside local address**—The IP address that is assigned to a host on the inside network. The address is probably not a legitimate IP address assigned by the Internet Network Information Center (InterNIC) or a service provider.
- **Inside global address**—A legitimate IP address (assigned by InterNIC or a service provider) that represents one or more inside local IP addresses to the outside world.
- **Outside local address**—The IP address of an outside host as it appears to the inside network. The address is not necessarily legitimate; it was allocated from the address space that is routable on the inside.
- **Outside global address**—The IP address that is assigned to a host on the outside network by the owner of the host. The address was allocated from a globally routable address or a network space.

**Pool Support for Dynamic NAT**

Cisco NX-OS provides pool support for dynamic NAT. Dynamic NAT allows the configuration of a pool of global addresses that can be used to dynamically allocate a global address from the pool for every new translation. The addresses are returned to the pool after the session ages out or is closed. This allows for a more efficient use of addresses based on requirements.
Support for PAT includes the use of the global address pool. This further optimizes IP address utilization. PAT exhausts one IP address at a time with the use of port numbers. If no port is available from the appropriate group and more than one IP address is configured, PAT moves to the next IP address and tries to allocate the original source port again. This process continues until PAT runs out of available ports and IP addresses.

With dynamic NAT and PAT, each host uses a different address or port for each subsequent translation. The main difference between dynamic NAT and static NAT is that static NAT allows a remote host to initiate a connection to a translated host if an access list exists that allows it, while dynamic NAT does not.

**Static and Dynamic Twice NAT Overview**

When both the source IP address and the destination IP address are translated as a single packet that goes through a Network Address Translation (NAT) device, it is referred to as twice NAT. Twice NAT is supported for static and dynamic translations.

Twice NAT allows you to configure two NAT translations (one inside and one outside) as part of a group of translations. These translations can be applied to a single packet as it flows through a NAT device. When you add two translations as part of a group, both the individual translations and the combined translation take effect.

A NAT inside translation modifies the source IP address and port number when a packet flows from inside to outside. It modifies the destination IP address and port number when the packet returns from outside to inside. NAT outside translation modifies the source IP address and port number when the packet flows from outside to inside, and it modifies the destination IP address and port number when the packet returns from inside to outside.

Without twice NAT, only one of the translation rules is applied on a packet, either the source IP address and port number or the destination IP address and port number.

Static NAT translations that belong to the same group are considered for twice NAT configuration. If a static configuration does not have a configured group ID, the twice NAT configuration will not work. All inside and outside NAT translations that belong to a single group that is identified by the group ID are paired to form twice NAT translations.

Dynamic twice NAT translations dynamically select the source IP address and port number information from pre-defined **ip nat pool** or **interface overload** configurations. Packet filtration is done by configuring ACLs, and traffic must originate from the dynamic NAT translation rule direction such that source translation is done by using dynamic NAT rules.

Dynamic twice NAT allows you to configure two NAT translations (one inside and one outside) as part of a group of translations. One translation must be dynamic and other translation must be static. When these two translations are part of a group of translations, both the translations can be applied on a single packet as it goes through the NAT device either from inside to outside or from outside to inside.

**VRF Aware NAT**

The VRF aware NAT feature enables a switch to understand an address space in a VRF (virtual routing and forwarding instances) and to translate the packet. This allows the NAT feature to translate traffic in an overlapping address space that is used between two VRFs.

Notes for VRF aware NAT:

- The VRF aware NAT feature is supported only on the Cisco Nexus 9300 platform switches.
The VRF aware NAT feature is not supported on the Cisco Nexus 9300-EX and 9300-FX platform switches.

This is a NAT TCAM limitation for the Cisco Nexus 9300-EX and 9300-FX platform switches. NAT TCAM is not VRF aware. NAT does not work with overlapping IP addresses on Cisco Nexus 9300-EX and 9300-FX platform switches.

Traffic flowing from one non-default-vrf to another non-default-vrf is not translated. (For example, vrfA to vrfB.)

For traffic flowing from a VRF to a global-VRF, a nat-outside configuration is not supported on a non-default VRF interface.

VRF aware NAT is supported by static and dynamic NAT configurations.

- When traffic is configured to flow from a non-default VRF (inside) to a default VRF (outside), the \texttt{match-in-vrf} option of the \texttt{ip nat} command cannot be specified.
- When traffic is configured to flow from a non-default VRF (inside) to the same non-default VRF (outside), the \texttt{match-in-vrf} option of the \texttt{ip nat} command must be specified.

The following is an example configuration:

```plaintext
Switch(config)#ip nat inside source list <ACL_NAME>
  <[interface <INTERFACE NAME> overload] | pool <POOL NAME> [overload]])
  [ group <1-1024> [dynamic] ] [ vrf <vrf-name> [match-in-vrf] ]
Switch(config)#ip nat inside source static [<LOCAL IP> <GLOBAL IP> | [tcp | udp] <LOCAL IP> <LOCAL PORT> <GLOBAL IP> <GLOBAL PORT>]
  [ group <1-1024> [dynamic] ] [ vrf <vrf-name> [match-in-vrf] ]

Switch(config)#ip nat outside source list <ACL_NAME>
  <[interface <INTERFACE NAME>] | pool <POOL NAME>)>
  [ group <1-1024> [dynamic] ] [ vrf <vrf-name> [match-in-vrf] ]
Switch(config)#ip nat outside source static [<LOCAL IP> <GLOBAL IP> | [tcp | udp] <LOCAL IP> <LOCAL PORT> <GLOBAL IP> <GLOBAL PORT>]
  [ group <1-1024> [dynamic] ] [ vrf <vrf-name> [match-in-vrf] ]
```

VRF aware NAT does not support fragmented packets.

VRF aware NAT does not support application layer translations.

Therefore, Layer 4 and other embedded IPs are not translated and the following will fail:

- FTP
- ICMP failures
- IPSec
- HTTPS
• VRF aware NAT supports NAT or VACL on an interface. (However, both features cannot be supported at the same time on an interface.)

• VRF aware NAT supports egress ACLs that are applied to the original packet, not on the NAT translated packet.

• VRF aware NAT supports only the default VRF.

• VRF aware NAT does not provide MIB support.

• VRF aware NAT does not provide DCNM support.

• VRF aware NAT supports only a single global VDC.

• VRF aware NAT does not support the active/standby supervisor model.

Guidelines and Limitations for Static NAT

Static NAT has the following configuration guidelines and limitations:

• For Broadcom-based Cisco Nexus 9000 Series switches, if the route to your inside global address on the translating device is reachable via the outside interface, packets for Network Address Translated flows coming from outside to inside get software forwarded, duplicated, and looped in the network. For this situation, you must enter the add-route CLI argument on the end of the NAT configuration for this flow. For example, `ip nat inside source static 192.168.1.1 172.16.1.1 add-route`.

• The static NAT feature over vPC is not supported on Cisco Nexus 9300 platform switches.

• `show` commands with the `internal` keyword are not supported.

• The static NAT feature is supported on Cisco Nexus 9300 platform switches.

• The static NAT feature is supported on Cisco Nexus 9200 platform switches.

• On Cisco Nexus 9200 and 9300-EX platform switches, the `add-route` option is required for both inside and outside policies.

Note

NAT is not supported on Cisco Nexus 9500 platform switches.

• NAT supports up to 1024 translations which include both static and dynamic NAT.

• If the translated IP is part of the outside interface subnet, then use the `ip proxy-arp` command on the NAT outside interface. If the `add-route` keyword is used, `ip proxy-arp` should be enabled.

• NAT and sFlow are not supported on the same port.

• The Cisco Nexus device supports NAT on the following interface types:
  • Switch Virtual Interfaces (SVIs)
  • Routed ports
  • Layer 3 and Layer 3 subinterfaces.
Restrictions for Dynamic NAT

The following restrictions apply to dynamic Network Address Translation (NAT):

- For Broadcom-based Cisco Nexus 9000 Series switches, if the route to your inside global address on the translating device is reachable via the outside interface, packets for Network Address Translated flows coming from outside to inside get software forwarded, duplicated, and looped in the network. For this situation, you must enter the `add-route` CLI argument on the end of the NAT configuration for this flow. For example, `ip nat inside source static 192.168.1.1 172.16.1.1 add-route`.

- VRF aware NAT is not supported for overlapping inside/outside IP subnet addresses on Cisco Nexus 9200 and 9300-EX platform switches.

- `show` commands with the `internal` keyword are not supported.

- The dynamic NAT feature is supported on Cisco Nexus 9300 platform switches.
• The dynamic NAT feature is supported on Cisco Nexus 9200 platform switches.

• On Cisco Nexus 9200 and 9300-EX platform switches, the `add-route` option is required for both inside and outside policies.

• The `interface overload option for inside policies` option is not supported on the Cisco Nexus 9200, 9300-EX, 9300-FX and 9300-FXP platform switches for both outside and inside policies.

• VXLAN routing is not supported on Cisco Nexus devices.

• Fragmented packets are not supported.

• Application layer gateway (ALG) translations are not supported. ALG, also known as application-level gateway, is an application that translates IP address information inside the payload of an application packet.

• Egress ACLs are not applied to translated packets.

• Nondefault virtual routing and forwarding (VRF) instances are not supported.

• MIBs are not supported.

• Cisco Data Center Network Manager (DCNM) is not supported.

• Multiple global virtual device contexts (VDCs) are not supported on Cisco Nexus devices.

• Dynamic NAT translations are not synchronized with active and standby devices.

• Stateful NAT is not supported. However, NAT and Hot Standby Router Protocol (HSRP) can coexist.

• The timeout value for take up to the configured time-out + 119 seconds.

• Normally, ICMP NAT flows time out after the expiration of the configured sampling-timeout and translation-timeout. However, when ICMP NAT flows present in the switch become idle, they time out immediately after the expiration of the sampling-timeout configured.

• Hardware programming is introduced for ICMP on Cisco Nexus 9300 platform switches. Therefore, the ICMP entries consume the TCAM resources in the hardware. Because ICMP is in the hardware, the maximum limit for NAT translation in Cisco Nexus platform Series switches is changed to 1024. Maximum of 100 ICMP entries are allowed to make the best usage of the resources.

• When creating a new translation on a Cisco Nexus 9000 Series switch, the flow is software forwarded until the translation is programmed in the hardware, which might take a few seconds. During this period, there is no translation entry for the inside global address. Therefore, returning traffic is dropped. To overcome this limitation, create a loopback interface and give it an IP address that belongs to the NAT pool.

• For dynamic NAT, pool overload and interface overload are not supported for the outside NAT.

• Because the NAT overload uses PBR (Policy-Based Routing), the maximum number of available next-hop entries in the PBR table determines NAT scale. If the number of NAT inside interfaces are within the range of available next-hops entries in the PBR table, the maximum NAT translation scale remains same. Otherwise, the maximum number of supported translations may reduce. PBR and NAT-overload are not mutually exclusive; they are mutually limiting.

• The Cisco Nexus devices does not support NAT and VLAN Access Control Lists (VACLs) that are configured on an interface at the same time.
Guidelines and Limitations for Dynamic Twice NAT

For Broadcom-based Cisco Nexus 9000 Series switches, if the route to your inside global address on the translating device is reachable via the outside interface, packets for Network Address Translated flows coming from outside to inside get software forwarded, duplicated, and looped in the network. For this situation, you must enter the `add-route` CLI argument on the end of the NAT configuration for this flow. For example, `ip nat inside source static 192.168.1.1 172.16.1.1 add-route`.

IP packets without TCP/UDP/ICMP headers are not translated with dynamic NAT.

In dynamic twice NAT, if dynamic NAT flows are not created before creating static NAT flows, dynamic twice NAT flows are not created correctly.

When an empty ACL is created, the default rule of `permit ip any any` is configured. The NAT-ACL does not match further ACL entries if the first ACL is blank.

Configuring Static NAT

Enabling Static NAT

**SUMMARY STEPS**

1. switch# configure terminal
2. switch(config)# feature nat
3. switch(config)# copy running-config startup-config

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enables the static NAT feature on the device.</td>
</tr>
<tr>
<td>switch(config)# feature nat</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.</td>
</tr>
<tr>
<td>switch(config)# copy running-config startup-config</td>
<td></td>
</tr>
</tbody>
</table>

Configuring Static NAT on an Interface

**SUMMARY STEPS**

1. switch# configure terminal
2. switch(config)# interface type slot/port
3. switch(config-if)# ip nat {inside | outside}
4. (Optional) switch(config)# copy running-config startup-config
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# interface type slot/port</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config-if)# ip nat {inside</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>(Optional) switch(config)# copy running-config startup-config</td>
</tr>
</tbody>
</table>

### Example
This example shows how to configure an interface with static NAT from the inside:

```plaintext
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# ip nat inside
```

### Enabling Static NAT for an Inside Source Address

For inside source translation, the traffic flows from inside interface to the outside interface. NAT translates the inside local IP address to the inside global IP address. On the return traffic, the destination inside global IP address gets translated back to the inside local IP address.

**Note**
When the Cisco Nexus device is configured to translate an inside source IP address (Src:ip1) to an outside source IP address (newSrc:ip2), the Cisco Nexus device implicitly adds a translation for an outside destination IP address (Dst: ip2) to an inside destination IP address (newDst: ip1).

### SUMMARY STEPS

1. switch# configure terminal
2. switch(config)# ip nat inside source static local-ip-address global-ip-address [group group-id] [vrf vrf-name [match-in-vrf]]
3. (Optional) switch(config)# copy running-config startup-config

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
</tbody>
</table>
Enabling Static NAT for an Outside Source Address

For outside source translation, the traffic flows from the outside interface to the inside interface. NAT translates the outside global IP address to the outside local IP address. On the return traffic, the destination outside local IP address gets translated back to outside global IP address.

### SUMMARY STEPS

1. `switch# configure terminal`
2. `switch(config)# ip nat outside source static global-ip-address [group group-id] [add-route] [vrf vrf-name [match-in-vrf]]`
3. (Optional) `switch(config)# copy running-config startup-config`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> switch(config)# ip nat outside source static global-ip-address [group group-id] [add-route] [vrf vrf-name [match-in-vrf]]</td>
<td>Configures static NAT to translate the outside global address to the outside local address or to translate the opposite (the outside local traffic to the outside global traffic). Specifying <code>group</code> specifies the group to which this translation belongs on the static twice NAT. When an inside translation without ports is configured, an implicit add route is performed. The</td>
</tr>
</tbody>
</table>
### Configuring Static and Dynamic NAT Translation

#### Configuring Static PAT for an Inside Source Address

You can map services to specific inside hosts using Port Address Translation (PAT).

**SUMMARY STEPS**

1. `switch# configure terminal`
2. `switch(config)# ip nat inside source static {inside-local-address outside-local-address | {tcp| udp} inside-local-address {local-tcp-port | local-udp-port} inside-global-address {global-tcp-port | global-udp-port} | group group-id | vrf vrf-name {match-in-vrf}}`
3. (Optional) `switch(config)# copy running-config startup-config`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# ip nat inside source static</td>
</tr>
<tr>
<td></td>
<td>{inside-local-address outside-local-address</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>(Optional) switch(config)# copy running-config startup-config</td>
</tr>
</tbody>
</table>

**Example**

This example shows how to configure static NAT for an outside source address:

```
switch# configure terminal
switch(config)# ip nat outside source static 2.2.2.2 6.6.6.6
switch(config)# copy running-config startup-config
```

This example shows how to map UDP services to a specific inside source address and UDP port:
### Configuring Static PAT for an Outside Source Address

You can map services to specific outside hosts using Port Address Translation (PAT).

#### SUMMARY STEPS

1. `switch# configure terminal`
2. `switch(config)# ip nat inside source static udp 20.1.9.2 63 35.48.35.48 130`
3. `(Optional) switch(config)# copy running-config startup-config`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>switch# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Maps static NAT to an outside global port to an outside local port.</td>
</tr>
<tr>
<td>`switch(config)# ip nat outside source static {outside-global-address outside-local-address</td>
<td>{tcp</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.</td>
</tr>
<tr>
<td><code>(Optional) switch(config)# copy running-config startup-config</code></td>
<td></td>
</tr>
</tbody>
</table>

#### Example

This example shows how to map TCP services to a specific outside source address and TCP port:

```
switch# configure terminal
switch(config)# ip nat outside source static tcp 20.1.9.2 63 35.48.35.48 130
switch(config)# copy running-config startup-config
```

### Configuring Static Twice NAT

All translations within the same group are considered for creating static twice Network Address Translation (NAT) rules.

#### SUMMARY STEPS

1. `enable`
2. configure terminal
3. ip nat inside source static inside-local-ip-address inside-global-ip-address [group group-id]
4. ip nat outside source static outside-global-ip-address outside-local-ip-address [group group-id] [add-route]
5. interface type number
6. ip address ip-address mask
7. ip nat inside
8. exit
9. interface type number
10. ip address ip-address mask
11. ip nat outside
12. end

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>enable&lt;br&gt;Example: switch&gt; enable</td>
<td>Enables privileged EXEC mode. &lt;br&gt;• Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2</td>
<td>configure terminal&lt;br&gt;Example: switch# configure terminal</td>
<td>Enters privileged EXEC mode.</td>
</tr>
<tr>
<td>Step 3</td>
<td>ip nat inside source static inside-local-ip-address inside-global-ip-address [group group-id]&lt;br&gt;Example: switch(config)# ip nat inside source static 10.1.1.1 192.168.34.4 group 4</td>
<td>Configures static twice NAT to translate an inside local IP address to the corresponding inside global IP address.&lt;br&gt;• The <code>group</code> keyword determines the group to which a translation belongs.</td>
</tr>
<tr>
<td>Step 4</td>
<td>ip nat outside source static outside-global-ip-address outside-local-ip-address [group group-id] [add-route]&lt;br&gt;Example: switch(config)# ip nat outside source static 209.165.201.1 10.3.2.42 group 4 add-route</td>
<td>Configures static twice NAT to translate an outside global IP address to the corresponding outside local IP address.&lt;br&gt;• The <code>group</code> keyword determines the group to which a translation belongs.</td>
</tr>
<tr>
<td>Step 5</td>
<td>interface type number&lt;br&gt;Example: switch(config)# interface ethernet 1/2</td>
<td>Configures an interface and enters interface configuration mode.</td>
</tr>
<tr>
<td>Step 6</td>
<td>ip address ip-address mask&lt;br&gt;Example: switch(config-if)# ip address 10.2.4.1 255.255.255.0</td>
<td>Sets a primary IP address for an interface.</td>
</tr>
<tr>
<td>Step 7</td>
<td>ip nat inside&lt;br&gt;Example:</td>
<td>Connects the interface to an inside network, which is subject to NAT.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>----------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)# ip nat inside</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Exit</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Exits interface configuration mode and returns to global configuration mode.</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)# exit</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>interface type number</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Configures an interface and enters interface configuration mode.</td>
<td></td>
</tr>
<tr>
<td>switch(config)# interface ethernet 1/1</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>ip address ip-address mask</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Sets a primary IP address for an interface.</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)# ip address 10.5.7.9 255.255.255.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>ip nat outside</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Connects the interface to an outside network, which is subject to NAT.</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)# ip nat outside</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td>end</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Exits interface configuration mode and returns to privileged EXEC mode.</td>
<td></td>
</tr>
<tr>
<td>switch(config-if)# end</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Enabling and Disabling no-alias Configuration**

NAT devices own Inside Global (IG) and Outside Local (OL) addresses and they are responsible for responding to any ARP requests directed to these addresses. When the IG/OL address subnet matches with the local interface subnet, NAT installs an IP alias and an ARP entry, in this case the device uses local-proxy-arp to respond to ARP requests.

The `no-alias` feature responds to ARP requests of all the translated IPs from a given NAT pool address range if the address range is in same subnet of the outside interface.

If no-alias is enabled on an interface with NAT configuration, the outside interface will not respond to any ARP requests in its subnet. When no-alias is disabled, the ARP requests for IPs in same subnet as of outside interface are served.

---

**Note** When you downgrade to any older releases that does not support this feature, configurations with `no-alias` option may be deleted.

**SUMMARY STEPS**

1. switch# configure terminal
2. switch(config)# feature nat
3. switch(config)# show run nat
4. `switch(config)# show ip nat-alias`
5. `switch(config)# clear ip nat-alias ip address/all`

### Detailed Steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>switch# configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>switch(config)# feature nat</code></td>
<td>Enables the static NAT feature on the device.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>switch(config)# show run nat</code></td>
<td>Displays NAT configuration.</td>
</tr>
<tr>
<td>Step 4</td>
<td><code>switch(config)# show ip nat-alias</code></td>
<td>Displays the information whether or not the alias is created. <strong>Note</strong> By default, alias is created. To disable the alias, you must append <em>no-alias</em> keyword to the command.</td>
</tr>
<tr>
<td>Step 5</td>
<td><code>switch(config)# clear ip nat-alias ip address/all</code></td>
<td>Removes entries from alias list. To remove a specific entry you must provide the IP address that you want to remove. To remove all entries, use the all keyword.</td>
</tr>
</tbody>
</table>

### Example

This example shows the interface information:

```
switch# configure terminal
switch(config)# show ip int b
IP Interface Status for VRF "default"  
<table>
<thead>
<tr>
<th>Interface</th>
<th>IP Address</th>
<th>Interface Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lo0</td>
<td>100.1.1.1</td>
<td>protocol-up/link-up/admin-up</td>
</tr>
<tr>
<td>Eth1/1</td>
<td>7.7.7.1</td>
<td>protocol-up/link-up/admin-up</td>
</tr>
<tr>
<td>Eth1/3</td>
<td>8.8.8.1</td>
<td>protocol-up/link-up/admin-up</td>
</tr>
</tbody>
</table>
```

This example shows the running configuration:

```
switch# configure terminal
switch(config)# show running-config nat
!Command: show running-config nat
!Running configuration last done at: Thu Aug 23 11:57:01 2018
!Time: Thu Aug 23 11:58:13 2018

version 9.2(2) Bios:version 07.64
feature nat
interface Ethernet1/1
  ip nat inside
interface Ethernet1/3
  ip nat outside
switch(config)#
```

This example shows how to configure alias:

```
switch# configure terminal
switch(config)# ip nat pool p1 7.7.7.2 7.7.7.20 prefix-length 24
switch(config)# ip nat inside source static 1.1.1.2 8.8.8.3
switch(config)# ip nat outside source static 2.2.2.1 7.7.7.3
switch(config)# show ip nat-alias
Alias Information for Context: default
```

---

Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide, Release 9.2(x)
This example shows the output of `show ip nat-alias`. By default, alias is enabled.

```
switch# configure terminal
switch(config)# show ip nat-alias
Alias Information for Context: default
Address Interface
7.7.7.2 Ethernet1/1
8.8.8.2 Ethernet1/3
```

This example shows how to disable alias:

```
switch# configure terminal
switch(config)# ip nat pool p1 7.7.7.2 7.7.7.20 prefix-length 24 no-alias
switch(config)# ip nat inside source static 1.1.1.2 8.8.8.3 no-alias
switch(config)# ip nat outside source static 2.2.2.1 7.7.7.3 no-alias
switch(config)# show ip nat-alias
Alias Information for Context: default
Address Interface
7.7.7.2 Ethernet1/1
8.8.8.2 Ethernet1/3
```

** None of the entry got appended as alias is disabled for above CLIs.

This example shows how to clear alias. Use `clear ip nat-alias` to remove an entry from alias list. You can remove a single entry by specifying the IP address or remove all the alias entries.

```
switch# configure terminal
switch(config)# clear ip nat-alias address 7.7.7.2
switch(config)# show ip nat-alias
Alias Information for Context: default
Address Interface
8.8.8.2 Ethernet1/3
```

### Configuration Example for Static NAT and PAT

This example shows the configuration for static NAT:

```
ip nat inside source static 103.1.1.1 11.3.1.1
ip nat inside source static 139.1.1.1 11.39.1.1
ip nat inside source static 141.1.1.1 11.41.1.1
ip nat inside source static 149.1.1.1 95.1.1.1
ip nat inside source static 149.2.1.1 96.1.1.1
ip nat outside source static 95.3.1.1 95.4.1.1
ip nat outside source static 96.3.1.1 96.4.1.1
ip nat outside source static 102.1.2.1 51.1.2.1
ip nat outside source static 104.1.1.1 51.3.1.1
ip nat outside source static 140.1.1.1 51.40.1.1
```

This example shows the configuration for static PAT:
Example: Configuring Static Twice NAT

The following example shows how to configure the inside source and outside source static twice NAT configurations:

Switch> enable
Switch# configure terminal
Switch(config)# ip nat inside source static 10.1.1.1 192.168.34.4 group 4
Switch(config)# ip nat outside source static 209.165.201.1 10.3.2.42 group 4
Switch(config)# interface ethernet 1/2
Switch(config-if)# ip address 10.2.4.1 255.255.255.0
Switch(config-if)# ip nat inside
switch(config-if)# exit
switch(config)# interface ethernet 1/1
switch(config-if)# ip address 10.5.7.9 255.255.255.0
switch(config-if)# ip nat outside
Switch(config-if)# end

Verifying the Static NAT Configuration

To display the static NAT configuration, perform this task:

SUMMARY STEPS

1. switch# show ip nat translations

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch# show ip nat translations</td>
<td>Shows the translations for the inside global, inside local, outside local, and outside global IP addresses.</td>
</tr>
</tbody>
</table>

Example

This example shows how to display the static NAT configuration:

switch# sh ip nat translations
Pro Inside global     Inside local     Outside local     Outside global
Configuring Dynamic NAT

Configuring Dynamic Translation and Translation Timeouts

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `ip access-list access-list-name`
4. `permit protocol source source-wildcard any`
5. `deny protocol source source-wildcard any`
6. `exit`
7. `ip nat inside source list access-list-name interface type number overload [vrf vrf-name [match-in-vrf]]`
8. `interface type number`
9. `ip address ip-address mask`
10. `ip nat inside`
11. `exit`
12. `interface type number`
13. ip address ip-address mask
14. ip nat outside
15. exit
16. ip nat translation max-entries number-of-entries
17. ip nat translation timeout seconds
18. ip nat translation creation-delay seconds
19. ip nat translation icmp-timeout seconds
20. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Switch&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ip access-list access-list-name</td>
<td>Defines an access list and enters access-list configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Switch(config)# ip access-list acl1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> permit protocol source source-wildcard any</td>
<td>Sets conditions in an IP access list that permit traffic matching the conditions.</td>
</tr>
<tr>
<td><strong>Example:</strong> Switch(config-acl)# permit ip 10.111.11.0/24 any</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> deny protocol source source-wildcard any</td>
<td>Sets conditions in an IP access list that deny packets from entering a network.</td>
</tr>
<tr>
<td><strong>Example:</strong> Switch(config-acl)# deny udp 10.111.11.100/32 any</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> exit</td>
<td>Exits access-list configuration mode and returns to global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Switch(config-acl)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> ip nat inside source list access-list-name interface type number overload [vrf vrf-name [match-in-vrf]]</td>
<td>Establishes dynamic source translation by specifying the access list defined in Step 3.</td>
</tr>
<tr>
<td><strong>Example:</strong> Switch(config)# ip nat inside source list acl1 interface ethernet 1/1 overload</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> interface type number</td>
<td>Configures an interface and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Switch(config)# interface ethernet 1/4</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>Sets a primary IP address for the interface.</td>
</tr>
<tr>
<td>ip address <em>ip-address mask</em></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Switch(config-if)# ip address 10.111.11.39 255.255.255.0</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>Connects the interface to an inside network, which is subject to NAT.</td>
</tr>
<tr>
<td>ip nat inside</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Switch(config-if)# ip nat inside</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>Exits interface configuration mode and returns to global configuration mode.</td>
</tr>
<tr>
<td>exit</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Switch(config-if)# exit</td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td>Configures an interface and enters interface configuration mode.</td>
</tr>
<tr>
<td>interface <em>type number</em></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Switch(config)# interface ethernet 1/1</td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td>Sets a primary IP address for an interface.</td>
</tr>
<tr>
<td>ip address <em>ip-address mask</em></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Switch(config-if)# ip address 172.16.232.182 255.255.255.240</td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td>Connects the interface to an outside network.</td>
</tr>
<tr>
<td>ip nat outside</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Switch(config-if)# ip nat outside</td>
</tr>
<tr>
<td><strong>Step 15</strong></td>
<td>Exits interface configuration mode and returns to global configuration mode.</td>
</tr>
<tr>
<td>exit</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Switch(config-if)# exit</td>
</tr>
<tr>
<td><strong>Step 16</strong></td>
<td>Specifies the maximum number of dynamic NAT translations. The number of entries can be between 1 and 1023.</td>
</tr>
<tr>
<td>ip nat translation max-entries <em>number-of-entries</em></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Switch(config)# ip nat translation max-entries 300</td>
</tr>
<tr>
<td><strong>Step 17</strong></td>
<td>Specifies the timeout value for dynamic NAT translations.</td>
</tr>
<tr>
<td>ip nat translation timeout <em>seconds</em></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Switch(config)# ip nat translation timeout 13000</td>
</tr>
<tr>
<td><strong>Step 18</strong></td>
<td>Specifies the ICMP timeout value for dynamic NAT translations.</td>
</tr>
<tr>
<td>ip nat translation creation-delay <em>seconds</em></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

**Command or Action**

```
switch(config)# ip nat translation creation-delay 250
```

**Purpose**

To reduce the frequency of programming the NAT entries in the hardware, NAT batches and programs the translations for one second. Frequently programming the hardware burdens the CPU but delaying the programming delays establishing sessions. You can disable batching or reduce the creation delay using this command. It is not recommended to set creation delay to 0.

**Note**

SpecifiestheICMPtimeoutvaluefordynamicNATtranslations.

### Configuring Dynamic NAT Pool

You can create a NAT pool by either defining the range of IP addresses in a single `ip nat pool` command or by using the `ip nat pool` and `address` commands.

#### SUMMARY STEPS

1. `switch# configure terminal`
2. `switch(config)# feature nat`
3. `switch(config)# ip nat pool pool-name [startip endip] {prefix prefix-length | netmask network-mask}`
4. (Optional) `switch(config-ipnat-pool)# address startip endip`
5. (Optional) `switch(config)# no ip nat pool pool-name`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>switch# configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>switch(config)# feature nat</code></td>
<td>Enables the NAT feature on the device.</td>
</tr>
<tr>
<td><strong>Step 3</strong> `switch(config)# ip nat pool pool-name [startip endip] {prefix prefix-length</td>
<td>netmask network-mask}`</td>
</tr>
<tr>
<td><strong>Step 4</strong> (Optional) <code>switch(config-ipnat-pool)# address startip endip</code></td>
<td>Specifies the range of global IP addresses if they were not specified during creation of the pool.</td>
</tr>
<tr>
<td><strong>Step 5</strong> (Optional) <code>switch(config)# no ip nat pool pool-name</code></td>
<td>Deletes the specified NAT pool.</td>
</tr>
</tbody>
</table>
Example
This example shows how to create a NAT pool with a prefix length:

```
switch# configure terminal
switch(config)# ip nat pool pool1 30.1.1.1 30.1.1.2 prefix-length 24
switch(config)#
```

This example shows how to create a NAT pool with a network mask:

```
switch# configure terminal
switch(config)# ip nat pool pool5 20.1.1.1 20.1.1.5 netmask 255.0.255.0
switch(config)#
```

This example shows how to create a NAT pool and define the range of global IP addresses using the `ip nat pool` and `address` commands:

```
switch# configure terminal
switch(config)# ip nat pool pool7 netmask 255.255.0.0
switch(config-ipnat-pool)# address 40.1.1.1 40.1.1.5
switch(config-ipnat-pool)#
```

This example shows how to delete a NAT pool:

```
switch# configure terminal
switch(config)# no ip nat pool pool4
switch(config)#
```

Configuring Source Lists

You can configure a source list of IP addresses for the inside interface and the outside interface.

Before you begin
Ensure that you configure a pool before configuring the source list for the pool.

SUMMARY STEPS

1. switch# configure terminal
2. (Optional) switch# ip nat inside source list list-name pool pool-name [overload]
3. (Optional) switch# ip nat outside source list list-name pool pool-name [add-route]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td>Step 2</td>
<td>(Optional) switch# ip nat inside source list list-name pool pool-name [overload]</td>
</tr>
</tbody>
</table>
### Configuring Dynamic Twice NAT for an Inside Source Address

For an inside source address translation, the traffic flows from the inside interface to the outside interface. You can configure dynamic twice NAT for an inside source address.

#### Before you begin

Ensure that you enable NAT on the switch.

#### SUMMARY STEPS

1. `switch# configure terminal`
2. `switch(config)# ip nat outside source static outside-global-ip-address outside-local-ip-address | [tcp | udp] outside-global-ip-address outside-global-port outside-local-ip-address outside-local-port [group group-id] [add-route] [dynamic]`
3. `switch(config)# ip nat inside source list access-list-name [interface type slot/port overload | pool pool-name ] [group group-id] [dynamic]`
4. `switch(config)# ip nat pool pool-name [startip endip] {prefix prefix-length | netmask network-mask}`
5. `switch(config)# interface type slot/port`
6. `switch(config-if)# ip nat outside`
7. `switch(config-if)# exit`
8. `switch(config)# interface type slot/port`
9. `switch(config-if)# ip nat inside`
DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# ip nat outside source static outside-global-ip-address outside-local-ip-address</td>
</tr>
<tr>
<td></td>
<td>outside-local-ip-address [tcp</td>
</tr>
<tr>
<td></td>
<td>udp] outside-global-ip-address outside-global-port</td>
</tr>
<tr>
<td></td>
<td>outside-local-ip-address outside-local-port [group</td>
</tr>
<tr>
<td></td>
<td>group-id] [add-route] [dynamic]</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config)# ip nat inside source list access-list-name</td>
</tr>
<tr>
<td></td>
<td>[interface type slot/port overload</td>
</tr>
<tr>
<td></td>
<td>pool pool-name ] [group group-id] [dynamic]</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>switch(config)# ip nat pool pool-name [startip endip] prefix prefix-length</td>
</tr>
<tr>
<td></td>
<td>netmask network-mask}</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>switch(config)# interface type slot/port</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>switch(config-if)# ip nat outside</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>switch(config-if)# exit</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>switch(config)# interface type slot/port</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>switch(config-if)# ip nat inside</td>
</tr>
</tbody>
</table>

**Example**

This example shows how to configure dynamic twice NAT for an inside source address:

```bash
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# ip nat outside source static 2.2.2.2 4.4.4.4 group 20 dynamic
switch(config)# ip nat inside source list acl_1 pool pool_1 overload group 20 dynamic
switch(config)# ip nat pool pool_1 3.3.3.3 3.3.3.10 prefix-length 24
switch(config)# interface Ethernet1/9
switch(config-if)# ip nat outside
switch(config-if)# exit
switch(config)# interface Ethernet1/15
switch(config-if)# ip nat inside
```
Configuring Dynamic Twice NAT for an Outside Source Address

For an outside source address translation, the traffic flows from the outside interface to the inside interface. You can configure dynamic twice NAT for an outside source address.

Before you begin

Ensure that you enable NAT on the switch.

SUMMARY STEPS

1. `switch# configure terminal`
2. `switch(config)# ip nat inside source static inside-local-ip-address inside-global-ip-address [tcp | udp] inside-local-ip-address local-port inside-global-ip-address global-port [group group-id] [dynamic]`
3. `switch(config)# ip nat outside source list access-list-name [interface type slot/port overload | pool pool-name] [group group-id] [add-route] [dynamic]`
4. `switch(config)# ip nat pool pool-name [startip endip] {prefix prefix-length | netmask network-mask}`
5. `switch(config)# interface type slot/port`
6. `switch(config-if)# ip nat outside`
7. `switch(config-if)# exit`
8. `switch(config)# interface type slot/port`
9. `switch(config-if)# ip nat inside`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config)# ip nat inside source static inside-local-ip-address inside-global-ip-address [tcp</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switch(config)# ip nat outside source list access-list-name [interface type slot/port overload</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>switch(config)# ip nat pool pool-name [startip endip] {prefix prefix-length</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>switch(config)# interface type slot/port</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>switch(config-if)# ip nat outside</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>switch(config-if)# exit</td>
</tr>
</tbody>
</table>
Configuring FINRST and SYN Timers

This section describes how to configure FINRST and SYN timer values. When you reload the switch, restoring or erasing the configured FINRST and/or SYN timer values depends on whether or not the TCP TCAM carved. If the TCAM is carved, the switch restores the currently configured values. If the timer values are not configured, it sets a default value of 60. If the TCAM is not carved, the switch removes any currently configured values and sets a default value as never. This is because the the TCP AWARE feature gets disabled when the TCP TCAM is not carved.

Before you begin

**SUMMARY STEPS**

1. switch# configure terminal
2. switch(config-if)# ip nat translation syn-timeout {seconds | never}
3. switch(config-if)# ip nat translation finrst-timeout {seconds | never}

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>switch(config-if)# ip nat translation syn-timeout {seconds</td>
</tr>
</tbody>
</table>
### Command or Action

| Step 3 | switch(config-if)# ip nat translation finrst-timeout {seconds | never} |

**Purpose**

- **Note**: You cannot configure SYN timer when TCP TCAM is not carved.

### Configuring Static and Dynamic NAT Translation

#### Clearing Dynamic NAT Translations

**To clear dynamic translations, perform the following task:**

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| clear ip nat translation [ all | inside
global-ip-address local-ip-address [outside local-ip-address | global-ip-address ] | outside local-ip-address global-ip-address ] | Deletes all or specific dynamic NAT translations. |

**Example**

This example shows how to clear all dynamic translations:

```bash
switch# clear ip nat translation all
```

This example shows how to clear dynamic translations for inside and outside addresses:

```bash
switch# clear ip nat translation inside 2.2.2.2 4.4.4.4 outside 5.5.5.5 7.7.7.7
```
Verifying Dynamic NAT Configuration

To display dynamic NAT configuration, perform the following tasks:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show ip nat translations</code></td>
<td>Displays active Network Address Translation (NAT) translations.</td>
</tr>
<tr>
<td></td>
<td>Displays additional information for each translation table entry, including when an entry was created and used.</td>
</tr>
<tr>
<td><code>show run nat</code></td>
<td>Displays NAT configuration.</td>
</tr>
</tbody>
</table>

Example

This example shows how to display running configuration for NAT:

```
switch# show run nat

!Command: show running-config nat

version 6.0(2)A3(1)
feature nat
ip nat inside source list list1 pool pool1
ip nat inside source list list2 pool pool2 overload
ip nat inside source list list7 pool pool7 overload
ip nat outside source list list3 pool pool13
ip nat pool pool1 30.1.1.1 30.1.1.2 prefix-length 24
ip nat pool pool2 10.1.1.1 10.1.1.2 netmask 255.0.255.0
ip nat pool pool3 30.1.1.1 30.1.1.8 prefix-length 24
ip nat pool pool5 20.1.1.1 20.1.1.5 netmask 255.0.255.0
ip nat pool pool7 netmask 255.255.0.0
    address 40.1.1.1 40.1.1.5
```

This example shows how to display active NAT translations:

Inside pool with overload

```
switch# show ip nat translation

Pro  Inside global  Inside local  Outside local  Outside global
icmp 20.1.1.3:64762 10.1.1.2:133 20.1.1.1:0 20.1.1.1:0
icmp 20.1.1.3:64763 10.1.1.2:134 20.1.1.1:0 20.1.1.1:0
```

Outside pool without overload

```
switch# show ip nat translation

Pro  Inside global  Inside local  Outside local  Outside global
any  ---  ---  177.7.1.1:0  77.7.1.64:0
any  ---  ---  40.146.1.1:0  40.46.1.64:0
any  ---  ---  10.4.146.1:0  10.4.46.64:0
```
Example: Configuring Dynamic Translation and Translation Timeouts

The following example shows how to configure dynamic overload Network Address Translation (NAT) by specifying an access list:

Switch> enable
Switch# configure terminal
Switch(config)# ip access-list acl1
Switch(config-acl)# permit ip 10.111.11.0/24 any
Switch(config-acl)# deny udp 10.111.11.100/32 any
Switch(config-acl)# exit
Switch(config)# ip nat inside source list acl1 interface ethernet 1/1 overload
Switch(config)# interface ethernet 1/4
Switch(config-if)# ip address 10.111.11.39 255.255.255.0
Switch(config-if)# ip nat inside
Switch(config-if)# exit
Switch(config)# interface ethernet 1/1
Switch(config-if)# ip address 172.16.232.182 255.255.255.240
Switch(config-if)# ip nat outside
Switch(config-if)# exit
Switch(config)# ip nat translation max-entries 300
Switch(config)# ip nat translation timeout 13000
Switch(config)# end
Example: Configuring Dynamic Translation and Translation Timeouts
CHAPTER 12

Configuring IP Event Dampening

• IP Event Dampening Overview, on page 369
• Guidelines and Limitations, on page 369
• Interface State Change Events, on page 370
• Affected Components, on page 371
• How to Configure IP Event Dampening, on page 372

IP Event Dampening Overview

Interface state changes occur when interfaces are administratively brought up or down or if an interface changes state. When an interface changes state or flaps, routing protocols are notified of the status of the routes that are affected by the change in state. Every interface state change requires all affected devices in the network to recalculate best paths, install or remove routes from the routing tables, and then advertise valid routes to peer routers. An unstable interface that flaps excessively can cause other devices in the network to consume substantial amounts of system processing resources and cause routing protocols to lose synchronization with the state of the flapping interface.

The IP Event Dampening feature introduces a configurable exponential decay mechanism to suppress the effects of excessive interface flapping events on routing protocols and routing tables in the network. This feature allows the network operator to configure a router to automatically identify and selectively dampen a local interface that is flapping. Dampening an interface removes the interface from the network until the interface stops flapping and becomes stable. Configuring the IP Event Dampening feature improves convergence times and stability throughout the network by isolating failures so that disturbances are not propagated. This, in turn, reduces the utilization of system processing resources by other devices in the network and improves overall network stability.

Guidelines and Limitations

The IP Event Dampening feature introduces a configurable exponential decay mechanism to suppress the effects of excessive interface flapping events on routing protocols and routing tables in the network. This feature allows the network operator to configure a router to automatically identify and selectively dampen a local interface that is flapping. See the following guidelines and limitations before configuring IP Event Dampening feature:

• Beginning from Cisco NX-OS Release 9.2(1), IP event dampening is supported on Cisco Nexus 9300-EX, 9300-FX, 9300-FX2, 9300-FXP, 9500-EX, and 9500-FX platform switches.
• Due to changes in the netstack-IP component, all the IP clients observe the impact of dampening or interface.

• For each flap of the interface, a certain penalty is added. The penalty decays exponentially whose parameters are configured.

• When penalty exceeds a certain high level, the interface is dampened. It is unsuppressed when the penalty decays below a low level.

• When an interface is dampened, the IP address and the static routes are removed from the interface. All the clients of IP get an IP delete notification.

• When an interface is unsuppressed, the IP address and the relevant routes are added back. All the clients of IP get an IP address add notification for all the IP addresses of the interface.

• All Layer 3 interfaces that are configured on the Ethernet interface, port channels, and SVI support this feature.

Interface State Change Events

IP Event Dampening feature employs a configurable exponential decay mechanism that is used to suppress the effects of excessive interface flapping or state changes. When the IP Event Dampening feature is enabled, flapping interfaces are dampened from the perspective of the routing protocol by filtering excessive route updates. Flapping interfaces are identified, assigned penalties, suppressed if necessary, and made available to the network when the interface stabilizes.

Suppress Threshold

The suppress threshold is the value of the accumulated penalty that triggers the router to dampen a flapping interface. The flapping interface is identified by the router and assigned a penalty for each up and down state change, but the interface is not automatically dampened. The router tracks the penalties that a flapping interface accumulates. When the accumulated penalty reaches the default or preconfigured suppress threshold, the interface is placed in a dampened state.

Half-Life Period

The half-life period determines how fast the accumulated penalty can decay exponentially. When an interface is placed in a dampened state, the router monitors the interface for additional up and down state changes. If the interface continues to accumulate penalties and the interface remains in the suppress threshold range, the interface will remain dampened. If the interface stabilizes and stops flapping, the penalty is reduced by half after each half-life period expires. The accumulated penalty will be reduced until the penalty drops to the reuse threshold. The configurable range of the half-life period timer is from 1 to 30 seconds. The default half-life period timer is 5 seconds.

Reuse Threshold

When the accumulated penalty decreases until the penalty drops to the reuse threshold, the route is unsuppressed and made available to other devices in the network. The range of the reuse value is from 1 to 20000 penalties. The default value is 1000 penalties.
### Maximum Suppress Time

The maximum suppress time represents the maximum time an interface can remain dampened when a penalty is assigned to an interface. The maximum suppress time can be configured from 1 to 255 seconds. The maximum penalty is truncated to maximum 20000 unit. The maximum value of the accumulated penalty is calculated based on the maximum suppress time, reuse threshold, and half-life period.

IP event dampening configuration command applies dampening to routing protocols for both IP and CLNS.

The first set of parameters ([half-life | reuse | suppress max-suppress]) configure the different parameters of the dampening algorithm. The second set ([restart [penalty ]]) enables dampening penalty to be applied when the interface comes up the first time after reboot. The default restart penalty is applied only if you specify the restart parameter. Both parameter sets are optional.

### Affected Components

When an interface is not configured with dampening, or when an interface is configured with dampening but is not suppressed, the routing protocol behavior as a result of interface state transitions is not changed by the IP Event Dampening feature. However, if an interface is suppressed, the routing protocols and routing tables are immune to any further state transitions of the interface until it is unsuppressed.

### Route Types

- **Connected routes:**
  - The connected routes of dampened interfaces are not installed into the routing table.
  - When a dampened interface is unsuppressed, the connected routes will be installed into the routing table if the interface is up.

- **Static routes:**
  - Static routes assigned to a dampened interface are not installed into the routing table.
  - When a dampened interface is unsuppressed, the static route will be installed into the routing table if the interface is up.

### Note

Only the primary interface can be configured with this feature, and all subinterfaces are subject to the same dampening configuration as the primary interface. IP Event Dampening does not track the flapping of individual subinterfaces on an interface.

### Supported Protocols

All the protocols that are used are impacted by the IP Event Dampening feature. The IP Event Dampening feature supports Border Gateway Protocol (BGP), Enhanced Interior Gateway Routing Protocol (EIGRP), Hot Standby Routing Protocol (HSRP), Open Shortest Path First (OSPF), Routing Information Protocol (RIP), and VRRP. Ping and SSH to the concerned interface IP address does not work.
The IP Event Dampening feature has no effect on any routing protocols if it is not enabled or an interface is not dampened.

How to Configure IP Event Dampening

Enabling IP Event Dampening

The `dampening` command is entered in interface configuration mode to enable the IP Event Dampening feature. If this command is applied to an interface that already has dampening configured, all dampening states are reset and the accumulated penalty will be set to 0. If the interface has been dampened, the accumulated penalty will fall into the reuse threshold range, and the dampened interface will be made available to the network. The flap counts, however, are retained.

SUMMARY STEPS

1. `configure terminal`
2. `interface type number`
3. `dampening [half-life-period reuse-threshold] [suppress-threshold max-suppress [restart-penalty]]`
4. `no dampening`
5. `end`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>interface type number</code></td>
<td>Enters interface configuration mode and configures the specified interface.</td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>dampening [half-life-period reuse-threshold] [suppress-threshold max-suppress [restart-penalty]]</code></td>
<td>Enables interface dampening.</td>
</tr>
<tr>
<td></td>
<td>• Entering the <code>dampening</code> command without any arguments enables interface dampening with default configuration parameters.</td>
</tr>
<tr>
<td></td>
<td>• When manually configuring the timer for the <code>restart-penalty</code> argument, the values must be manually entered for all arguments.</td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>no dampening</code></td>
<td>Disables interface dampening.</td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>end</code></td>
<td>Exits interface configuration mode.</td>
</tr>
</tbody>
</table>
Verifying IP Event Dampening

Use the `show dampening interface` or `show interface dampening` commands to verify the configuration of the IP Event Dampening feature.

**SUMMARY STEPS**

1. `show ip interface [interface]`
2. `show dampening interface`
3. `show interface dampening`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>show ip interface [interface]</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>show dampening interface</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>show interface dampening</code></td>
</tr>
</tbody>
</table>

**Default Settings for IP Dampening Parameters**

*Table 16: Default values for IP Dampening Parameters*

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Range</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Half-life</td>
<td>1-30</td>
<td>5</td>
</tr>
<tr>
<td>Reuse threshold</td>
<td>1-20000</td>
<td>800</td>
</tr>
<tr>
<td>Suppress threshold</td>
<td>1-20000</td>
<td>2000</td>
</tr>
<tr>
<td>Max suppress time</td>
<td>1-255 seconds</td>
<td>20 seconds</td>
</tr>
<tr>
<td>Apply restart penalty</td>
<td></td>
<td>False</td>
</tr>
<tr>
<td>Restart penalty</td>
<td>true / false</td>
<td>false</td>
</tr>
</tbody>
</table>
Default Settings for IP Dampening Parameters
Configuring IP TCP MSS

- Information About IP TCP MSS, on page 375
- Licensing Requirements for IP TCP MSS, on page 375
- Default Settings for IP TCP MSS, on page 376
- Guidelines and Limitations for IP TCP MSS, on page 376
- Configuring IP TCP MSS, on page 376
- Verifying IP TCP MSS, on page 378

Information About IP TCP MSS

The IP TCP Maximum Segment Size (MSS) feature enables a switch to set a maximum segment size for all TCP connections that originate or terminate at a Cisco Nexus 9000 Series switch. The MSS in a TCP header field is the maximum data size or payload that a host can send or receive in a single segment. By default, a Cisco Nexus 9000 Series switch sets the MSS value to 536 bytes for IPv4 TCP connections and 1240 bytes for IPv6 TCP connections. This default value is set by the switch during the initial TCP connection establishment.

The switch from which the TCP connection originates will always set the MSS to the user-configured MSS or the difference between the route interface MTU and the protocol header, whichever is lower. Thus, Host A sends a SYN packet with the proposed MSS of 1460 bytes to Host B. After receiving the SYN packet with the proposed MSS, Host B sends a SYN-ACK packet to Host A, accepting the proposed MSS value for the TCP connection. Host A sends an ACK packet to Host B, setting the MSS value to 1460 for the TCP connection.

Licensing Requirements for IP TCP MSS

IP TCP MSS requires no license. Any feature that is not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing scheme, see the Cisco NX-OS Licensing Guide.
Default Settings for IP TCP MSS

Table 17: Default Settings for IP TCP MSS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP TCP MSS</td>
<td>536 bytes for IPv4 TCP connections</td>
</tr>
<tr>
<td></td>
<td>1240 bytes for IPv6 TCP connections</td>
</tr>
</tbody>
</table>

Guidelines and Limitations for IP TCP MSS

If the MSS has to be set to a value that is more than 1460 bytes for IPv4 TCP connections, the corresponding MTU value should be set to the required MSS value plus 40 bytes. If the MSS has to be set to a value that is more than 1440 bytes for IPv6 TCP connections, the corresponding MTU value should be set to the required MSS value plus 60 bytes.

Configuring IP TCP MSS

Setting the MSS for TCP Connections, on page 376
Removing a Set IP TCP MSS, on page 377

Setting the MSS for TCP Connections

Before you begin

SUMMARY STEPS

1. switch# configure terminal
2. switch(config)# ip tcp <bytes>
3. switch# show ip tcp mss

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 switch# configure terminal</td>
<td>Enter global configuration mode</td>
</tr>
<tr>
<td>Step 2 switch(config)# ip tcp &lt;bytes&gt;</td>
<td>Set a maximum segment size.</td>
</tr>
<tr>
<td>Step 3 switch# show ip tcp mss</td>
<td>Display the configured IP TCP MSS.</td>
</tr>
</tbody>
</table>

Example: Running Configuration
Example

This example shows a running configuration, followed by a verification command that displays the configured IP TCP MSS:

```config
configure terminal
ip tcp mss 5000
Setting TCP MSS to 5000 bytes

switch# show ip tcp mss
TCP MSS value 5000 bytes
```

Removing a Set IP TCP MSS

**SUMMARY STEPS**

1. `switch# configure terminal`
2. `switch(config)# no ip tcp mss`
3. `switch# show ip tcp mss`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enter global configuration mode</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Remove the configured IP TCP MSS and set the IP TCP MSS to default values.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Display the configured IP TCP MSS.</td>
</tr>
</tbody>
</table>

Example: Running Configuration

Example

This example shows a running configuration, followed by a verification command that displays the configured IP TCP MSS:

```config
configure terminal
no ip tcp mss 5000
Setting default MSS value is 536 bytes

switch# show ip tcp mss
TCP MSS value 536 bytes
```

Example: Setting the MSS for TCP Connections

This example shows a setting the MSS for TCP connections:

```config
configure terminal
ip tcp mss 2000
```
Example: Removing a Set IP TCP MSS

This example shows how to remove the MSS:

```
configure terminal
no ip tcp mss
```

Verifying IP TCP MSS

Table 18: Verifying IP TCP MSS

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>show ip tcp mss</td>
<td>Displays the set IP TCP MSS.</td>
</tr>
</tbody>
</table>
Configuring Layer 2 Data Center Interconnect

This section contains an example of how to configure a Layer 2 Data Center Interconnect (DCI) with the use of a Virtual Port-Channel (vPC).

- **Overview, on page 379**
- **Example of Layer 2 Data Center Interconnect, on page 379**

## Overview

The purpose of a Data Center Interconnect (DCI) is to extend specific VLANs between different data centers, which offers Layer 2 adjacency for servers and Network Attached Storage (NAS) devices that are separated by large distances.

The vPC provides the benefit of STP isolation between the two sites (no Bridge Protocol Data Unit (BPDU) across the DCI vPC) This means that an outage in the data center is not propagated to the remote data center because redundant links are still provided between the data centers.

### Note

The vPC can be used to interconnect a maximum of two data centers.

### Note

The supported platforms include Cisco Nexus 9500 Series switches with N9K-X9636C-R, N9K-X9636Q-R, N9K-X9636C-RX line cards.

## Example of Layer 2 Data Center Interconnect

The following is an example configuration of a Layer 2 Data Center Interconnect (DCI) with use of vPC. The example allows for First Hop Redundancy Protocol (FHRP) isolation.

### Note

vPC and Hot Standby Routing Protocol (HSRP) have already been configured.
Link Aggregation Control Protocol (LACP) should be used on the vPC link, which acts as the DCI.

**Note**

Figure 39: Dual Layer 2/Layer 3 POD Interconnect

In this example, the Layer 3 (L3) gateway is configured on the same vPC pair and acts as the DCI. In order to isolate the Hot Standby Routing Protocol (HSRP), you must configure a Port Access Control List (PACL) on the DCI port-channel and disable HSRP Gratuitous Address Resolution Protocols (ARPs) (GARPs) on the Switched Virtual Interfaces (SVIs) for the VLANs that move across the DCI.

```
ip access-list DENY_HSRP_IP
   10 deny udp any 224.0.0.2/32 eq 1985
   20 deny udp any 224.0.0.102/32 eq 1985
   30 permit ip any any

interface <DCI-Port-Channel>
   ip port access-group DENY_HSRP_IP in

interface Vlan <x>
   no ip arp gratuitous hsrp duplicate
```
IETF RFCs supported by Cisco NX-OS Interfaces

This appendix lists the IETF RFCs for interfaces supported by Cisco NX-OS.

- IPv6 RFCs, on page 381

### IPv6 RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1981 (7.0(3)I1(1) and later)</td>
<td>Path MTU Discovery for IP version 6</td>
</tr>
<tr>
<td>RFC 2373</td>
<td>IPv Version 6 Addressing Architecture</td>
</tr>
<tr>
<td>RFC 2374</td>
<td>An Aggregatable Global Unicast Address Format</td>
</tr>
<tr>
<td>RFC 2460</td>
<td>Internet Protocol, Version 6 (IPv6) Specification</td>
</tr>
<tr>
<td>RFC 2462</td>
<td>IPv6 Stateless Address Autoconfiguration</td>
</tr>
<tr>
<td>RFC 2464</td>
<td>Transmission of IPv6 Packets over Ethernet Networks</td>
</tr>
<tr>
<td>RFC 2467</td>
<td>Transmission of IPv6 Packets over FDDI Networks</td>
</tr>
<tr>
<td>RFC 2472</td>
<td>IPv6 over ATM Networks</td>
</tr>
<tr>
<td>RFC 2492</td>
<td>IPv6 over ATM Networks</td>
</tr>
<tr>
<td>RFC 2590</td>
<td>Transmission of IPv6 Packets over Frame Relay Networks Specification</td>
</tr>
<tr>
<td>RFC 3021</td>
<td>Using 31-Bit Prefixes on IPv4 Point-to-Point Links</td>
</tr>
<tr>
<td>RFC 3152</td>
<td>Delegation of IP6.ARPA</td>
</tr>
<tr>
<td>RFC 3162</td>
<td>RADIUS and IPv6</td>
</tr>
<tr>
<td>RFC 3513</td>
<td>Internet Protocol Version 6 (IPv6) Addressing Architecture</td>
</tr>
<tr>
<td>RFC 3596</td>
<td>DNS Extensions to Support IP version 6</td>
</tr>
<tr>
<td>RFC 4193</td>
<td>Unique Local IPv6 Unicast Addresses</td>
</tr>
</tbody>
</table>
APPENDIX C

Configuration Limits for Cisco NX-OS Interfaces

The configuration limits are documented in the *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*. 
INDEX

A

address 359, 360
admin-shutdown 97
auto-recovery 247, 277, 280
autonomous-system 148

duplex half 193, 194

E

enable 350, 351, 356, 357
encapsulation dot1Q 100, 101, 102, 107, 108, 109, 110
end 109, 110, 357, 359
errdisable detect cause 12, 27
errdisable detect cause acl-exception 27
erd disable detect cause all 27
erd disable detect cause link-flap 27
erd disable detect cause loopback 27
erd disable recovery cause 12, 28, 29
erd disable recovery cause all 28, 29
erd disable recovery cause bduguard 28, 29
erd disable recovery cause failed-port-state 28, 29
erd disable recovery cause link-flap 28, 29
erd disable recovery cause loopback 28, 29
erd disable recovery cause misabling 28, 29
erd disable recovery cause psecure-violation 28, 29
erd disable recovery cause security-violation 28, 29
erd disable recovery cause storm-control 28, 29
erd disable recovery cause udl 28, 29
erd disable recovery cause vpc-peerlink 28, 29
erd disable recovery interval 12, 29, 30
ethernet 24
Ethernet 5

F

feature bfd 132
feature eigrp 36
feature interface-vlan 82, 103, 104
feature isis 109
feature lacp 196
feature nat 346, 359
feature tunnel 302
feature vpc 254

graceful consistency-check 268, 269

G

delay 35, 189, 190
delay restore 241, 248
deny 356, 357
description 24, 192, 193, 303
dual-active exclude interface-vlan 226
duplex 193, 194
duplex auto 193, 194
duplex full 193, 194

INDEX

A

address 359, 360
admin-shutdown 97
auto-recovery 247, 277, 280
autonomous-system 148

duplex half 193, 194

E

enable 350, 351, 356, 357
encapsulation dot1Q 100, 101, 102, 107, 108, 109, 110
end 109, 110, 357, 359
errdisable detect cause 12, 27
errdisable detect cause acl-exception 27
erd disable detect cause all 27
erd disable detect cause link-flap 27
erd disable detect cause loopback 27
erd disable recovery cause 12, 28, 29
erd disable recovery cause all 28, 29
erd disable recovery cause bduguard 28, 29
erd disable recovery cause failed-port-state 28, 29
erd disable recovery cause link-flap 28, 29
erd disable recovery cause loopback 28, 29
erd disable recovery cause misabling 28, 29
erd disable recovery cause psecure-violation 28, 29
erd disable recovery cause security-violation 28, 29
erd disable recovery cause storm-control 28, 29
erd disable recovery cause udl 28, 29
erd disable recovery cause vpc-peerlink 28, 29
erd disable recovery interval 12, 29, 30
ethernet 24
Ethernet 5

F

feature bfd 132
feature eigrp 36
feature interface-vlan 82, 103, 104
feature isis 109
feature lacp 196
feature nat 346, 359
feature tunnel 302
feature vpc 254

graceful consistency-check 268, 269

G

delay 35, 189, 190
delay restore 241, 248
deny 356, 357
description 24, 192, 193, 303
dual-active exclude interface-vlan 226
duplex 193, 194
duplex auto 193, 194
duplex full 193, 194

INDEX

A

address 359, 360
admin-shutdown 97
auto-recovery 247, 277, 280
autonomous-system 148

duplex half 193, 194

E

enable 350, 351, 356, 357
encapsulation dot1Q 100, 101, 102, 107, 108, 109, 110
end 109, 110, 357, 359
errdisable detect cause 12, 27
errdisable detect cause acl-exception 27
erd disable detect cause all 27
erd disable detect cause link-flap 27
erd disable detect cause loopback 27
erd disable recovery cause 12, 28, 29
erd disable recovery cause all 28, 29
erd disable recovery cause bduguard 28, 29
erd disable recovery cause failed-port-state 28, 29
erd disable recovery cause link-flap 28, 29
erd disable recovery cause loopback 28, 29
erd disable recovery cause misabling 28, 29
erd disable recovery cause psecure-violation 28, 29
erd disable recovery cause security-violation 28, 29
erd disable recovery cause storm-control 28, 29
erd disable recovery cause udl 28, 29
erd disable recovery cause vpc-peerlink 28, 29
erd disable recovery interval 12, 29, 30
ethernet 24
Ethernet 5

F

feature bfd 132
feature eigrp 36
feature interface-vlan 82, 103, 104
feature isis 109
feature lacp 196
feature nat 346, 359
feature tunnel 302
feature vpc 254

graceful consistency-check 268, 269

G

delay 35, 189, 190
delay restore 241, 248
deny 356, 357
description 24, 192, 193, 303
dual-active exclude interface-vlan 226
duplex 193, 194
duplex auto 193, 194
duplex full 193, 194
INDEX

H
hardware access-list team region nat 344
how l2protocol tunnel summary 335
hsrp bfd 152
hsrp bfd all-interfaces 152

I
iip nat inside source list 360
iip nat inside source static 363
include bfd 132
interface breakout 5
interface ether 52
interface ethernet 26, 31, 34, 35, 39, 41, 66, 68, 72, 73, 77, 98, 100, 106, 107, 108, 109, 110, 118, 323, 327, 328, 329, 330, 332
interface loopback 105, 106, 107, 109
interface overload 341
interface port-channel 72, 73, 77, 79, 102, 136, 157, 158, 159, 184, 189, 190, 191, 192, 193, 194, 198, 199, 200, 205, 206, 207, 208, 209, 210, 213, 260, 270, 271
interface tunnel 303, 304, 305, 306, 307, 308, 310, 311, 312
interface vlan 82, 103, 104
interfaces-vlan 241, 248
ip 20
ip address 356, 357
ip address dhcp 96
ip address dhcp 96
ip address use-link-local-only 96
ip ospf authentication 19, 49, 50, 51
ip ospf authentication-key 107, 108
ip ospf bfd 149, 157, 158, 159, 160
ip ospf bfd disable 156
ip pim bfd 154
ip pim bfd-instance 154, 155
ip pim pre-build-spt 240
ip pim spt-threshold infinity 239
ip pim use-shared-tree-only 239
ip route 96, 155, 156
ip route static bfd 155, 156
ip router isis 109, 110
ip router ospf 107, 108, 109
ip unnumbered 107, 108, 109, 110
ipv6 address 98, 99, 100, 101, 102, 103, 104, 105, 106, 116
ipv6 address dhcp 96
ipv6 address use-link-local-only 96
ipv6 nd mac-extract 116
ipv6 nd synchronize 238
isis bfd 150, 151
isis bfd disable 156

L
l2protocol tunnel 330
l2protocol tunnel cos 331
l2protocol tunnel drop-threshold 332
l2protocol tunnel shutdown-threshold 332, 333
lacp graceful-convergence 179, 206, 207
lacp max-bundle 200
lacp min-links 198, 199
lacp mode delay 210
lacp port-priority 203
lacp rate 200
lacp rate fast 201
lacp suspend-individual 179, 207, 209
lacp system-priority 202
link debounce link-up 41, 42
link debounce time 41
load- interval 88, 120, 215, 216
load-interval counters 52
loopback 107, 108, 110

M
mac-address ipv6-extract 116
match-in-vrf 342
medium 99
medium broadcast 99
medium p2p 99, 107, 108, 109, 110
mgmt0 24
mtu 31, 32, 33, 303, 307, 308

N
negotiate auto 19, 49, 50, 51
negotiate auto 25000 49
neighbor 146, 163
net 109, 110

P
p2p 99
peer-gateway 227, 261, 262
peer-gateway exclude-vlan 227
peer-keepalive destination 258
peer-switch 284, 286
permit 356, 357
permit ip any any 346
port-channel load-balance 172, 195

R
regex 20
reload restore 277, 278
role priority 275
router bgp 146, 163
router eigrp 147, 148
router isis 109, 110, 150
router ospf 149

S
show 99
show bfd 161
show bfd neighbors 161
show cdp all 51
show cfs application 247
show dot1q-tunnel 323, 335
show feature 132, 215, 254, 255, 256, 290, 302
show hardware feature-capability 220
show hsrp detail 151
show interface 24, 25, 36, 37, 51, 52, 53, 66, 67, 68, 69, 70, 78, 186, 187, 188, 189
show interface brief 51, 86, 87
show interface capabilities 87
show interface counters 88, 216
show interface counters detailed 88, 216
show interface counters errors 88, 216
show interface eth 25, 101
show interface ethernet 26, 31, 34, 35, 87, 119, 120
show interface ethernet errors 121
show interface loopback 105, 106, 120, 121
show interface port-channel 120, 121, 190, 191, 192, 193, 194, 215
show interface status err-disabled 12, 27, 28, 29, 30, 51
show interface switchport 87
show interface transceivers 19
show interface trunk 87
show interface tunnel 313
show interface vlan 103, 104, 120, 121
show interfaces 100, 101
show interfaces tunnel 304, 307, 308, 310
show ip copy 20
show ip eigrp 147, 148
show ip interface brief 120
show ip nat translations 355, 366
show ip ospf 149
show ip route 120
show ip route static 155, 156
show ipv6 icmp interface 116
show isis 150, 151
show l2protocol tunnel 335
show lacp 216
show lacp counters 216
show lacp system-identifier 202
show mac address-table 247
show port-channel capacity 290
show port-channel compatibility-parameters 170, 215
show port-channel database 215
show port-channel load-balance 195, 196, 216
show port-channel summary 184, 185, 197, 216
show port-channel traffic 216
show port-channel usage 216
show run nat 366
show running config 99
show running-config 81, 87
show running-config all 32, 33
show running-config bfd 133, 134, 135, 136, 137, 138, 161
show running-config bgp 146, 147
show running-config hsrp 152
show running-config interface ethernet 87
show running-config interface port-channel 77, 79, 80, 87, 200
show running-config interface vlan 82, 87
show running-config l2pt 335
show running-config pim 154, 155
show running-config vpc 278, 280, 290
show running-config vrrp 153, 154
show spanning-tree 237
show spanning-tree summary 284, 285, 286, 287
show startup-config bfd 161
show startup-config interface vlan 82, 83
show udlld 39, 40, 51
show udlld global 51
show vlan 71, 72, 73
show vpc brief 232, 237, 257, 258, 260, 261, 262, 268, 269, 270, 271, 276, 277, 290
show vpc consistency-parameters 230, 231, 232, 269, 270, 290
show vpc consistency-parameters global 269, 270
show vpc consistency-parameters interface port-channel 269, 270, 278, 280
show vpc orphan-ports 281, 282
show vpc peer-keepalive 290
show vpc role 272, 273, 274, 275, 290
show vpc statistics 258, 259, 290
show vrf 114, 115, 312
show vrrp detail 153
shutdown 12, 27, 28, 36, 37, 191, 205, 206, 207, 208, 209, 224
spanning-tree port type edge 322
spanning-tree pseudo-information 285, 286
spanning-tree vlan 284
speed 193, 194
speed 10 193, 194
speed 100 193, 194
speed 1000 193, 194
speed auto 19, 193, 194
<table>
<thead>
<tr>
<th>Speed Group 54</th>
<th>Speed Group 10000 18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static 338</td>
<td></td>
</tr>
<tr>
<td>Switchport 18, 31, 56, 79, 99, 100, 186, 323, 327, 328, 329, 330, 332</td>
<td></td>
</tr>
<tr>
<td>Switchport Access VLAN 66, 67, 322</td>
<td></td>
</tr>
<tr>
<td>Switchport Dot1Q Ethertype 327, 328, 329</td>
<td></td>
</tr>
<tr>
<td>Switchport Host 68</td>
<td></td>
</tr>
<tr>
<td>Switchport Isolated 77</td>
<td></td>
</tr>
<tr>
<td>Switchport Mode 62, 66, 69, 70, 329</td>
<td></td>
</tr>
<tr>
<td>Switchport Mode Dot1Q-Tunnel 323, 330, 332</td>
<td></td>
</tr>
<tr>
<td>Switchport Mode Trunk 184, 186, 260</td>
<td></td>
</tr>
<tr>
<td>Switchport Trunk 186</td>
<td></td>
</tr>
<tr>
<td>Switchport Trunk Allowed VLAN 70, 72, 73, 186, 260</td>
<td></td>
</tr>
<tr>
<td>Switchport Trunk Native 186</td>
<td></td>
</tr>
<tr>
<td>Switchport Trunk Native VLAN 71</td>
<td></td>
</tr>
<tr>
<td>System Default Interface-VLAN Autostate 81</td>
<td></td>
</tr>
<tr>
<td>System Default Switchport 56, 85, 96</td>
<td></td>
</tr>
<tr>
<td>System Default Switchport Shutdown 86</td>
<td></td>
</tr>
<tr>
<td>System Jumbo MTU 32</td>
<td></td>
</tr>
<tr>
<td>System MAC 272</td>
<td></td>
</tr>
<tr>
<td>System Priority 273, 274</td>
<td></td>
</tr>
<tr>
<td>Tunnel Destination 304, 305, 306</td>
<td></td>
</tr>
<tr>
<td>Tunnel Mode 303, 304, 307, 308</td>
<td></td>
</tr>
<tr>
<td>Tunnel Mode GRE IP 303, 304, 310</td>
<td></td>
</tr>
<tr>
<td>Tunnel Mode IPIP 303, 304, 305, 306, 307, 308</td>
<td></td>
</tr>
<tr>
<td>Tunnel Mode IPv6IPv6 Decapsulate-Any 307, 308</td>
<td></td>
</tr>
<tr>
<td>Tunnel Path-MTU Discovery 311</td>
<td></td>
</tr>
<tr>
<td>Tunnel Path-MTU Discovery Age-Timer 311</td>
<td></td>
</tr>
<tr>
<td>Tunnel Path-MTU Discovery Min-MTU 311</td>
<td></td>
</tr>
<tr>
<td>Tunnel Source 303, 304, 305, 306</td>
<td></td>
</tr>
<tr>
<td>Tunnel TTL 303</td>
<td></td>
</tr>
<tr>
<td>Tunnel Use-VRF 304</td>
<td></td>
</tr>
<tr>
<td>U</td>
<td></td>
</tr>
<tr>
<td>UDLD 39</td>
<td></td>
</tr>
<tr>
<td>UDLD Aggressive 39</td>
<td></td>
</tr>
<tr>
<td>UDLD Message-Time 39</td>
<td></td>
</tr>
<tr>
<td>Update-Source 146, 147, 163, 164</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
</tr>
<tr>
<td>VLAN 286</td>
<td></td>
</tr>
<tr>
<td>VLAN Dot1Q Tag Native 318</td>
<td></td>
</tr>
<tr>
<td>VPC 270, 271</td>
<td></td>
</tr>
<tr>
<td>VPC Domain 256, 257, 258, 261, 268, 272, 273, 274, 275, 276, 277, 278, 280, 284, 286</td>
<td></td>
</tr>
<tr>
<td>VPC Orphan-Ports Suspend 252, 281, 282</td>
<td></td>
</tr>
<tr>
<td>VPC Peer-Link 260</td>
<td></td>
</tr>
<tr>
<td>VRF Context 155</td>
<td></td>
</tr>
<tr>
<td>VRF Member 114, 115, 311, 312</td>
<td></td>
</tr>
<tr>
<td>VRRP 153</td>
<td></td>
</tr>
<tr>
<td>VRRP BFD 153</td>
<td></td>
</tr>
</tbody>
</table>