



Cisco Nexus 3000 Series NX-OS Unicast Routing Configuration Guide, Release 6.x

First Published: May 2013

Last Modified: February 2016

Cisco Systems, Inc.
www.cisco.com

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

Text Part Number: OL-29560-03

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.

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: 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. (1721R)

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

Cisco Nexus 3000 Series NX-OS Unicast Routing Configuration Guide, Release 6.x
©2016 Cisco Systems, Inc. All rights reserved.



Preface xxiii

Audience xxiii

Document Conventions xxiii

Obtaining Documentation and Submitting a Service Request 3-xxiv

3-xxiv

New and Changed Information 1

CHAPTER 1

Overview 1-1

Information About Layer 3 Unicast Routing 1-1

Routing Fundamentals 1-2

Packet Switching 1-2

Routing Metrics 1-3

Path Length 1-4

Reliability 1-4

Routing Delay 1-4

Bandwidth 1-4

Load 1-4

Communication Cost 1-4

Router IDs 1-5

Autonomous Systems 1-5

Convergence 1-6

Load Balancing and Equal Cost Multipath 1-6

Route Redistribution 1-6

Administrative Distance 1-7

Stub Routing 1-7

Routing Algorithms 1-8

Static Routes and Dynamic Routing Protocols 1-8

Interior and Exterior Gateway Protocols 1-8

Distance Vector Protocols 1-9

Link-State Protocols 1-9

Layer 3 Virtualization 1-10

Cisco NX-OS Forwarding Architecture 1-10

Unicast RIB	1-10
Adjacency Manager	1-11
Unicast Forwarding Distribution Module	1-11
FIB	1-12
Unified Forwarding Table	1-12
Hardware Forwarding	1-12
Software Forwarding	1-13
Summary of Layer 3 Unicast Routing Features	1-13
OSPF	1-13
EIGRP	1-13
BGP	1-13
RIP	1-14
Static Routing	1-14
Layer 3 Virtualization	1-14
Route Policy Manager	1-14
First-Hop Redundancy Protocols	1-14
Object Tracking	1-14
Related Topics	1-15

CHAPTER 2
Configuring IPv4 2-1

Information About IPv4	2-1
Multiple IPv4 Addresses	2-2
Address Resolution Protocol	2-3
ARP Caching	2-3
Static and Dynamic Entries in the ARP Cache	2-3
Devices That Do Not Use ARP	2-4
Reverse ARP	2-4
Proxy ARP	2-5
Local Proxy ARP	2-5
Gratuitous ARP	2-5
Glean Throttling	2-5
ICMP	2-6
ICMP Unreachable Support to Set Source Interface	2-6
Virtualization Support	2-6
IPv4 Routes with ECMP	2-6
Licensing Requirements for IPv4	2-7
Prerequisites for IPv4	2-7
Guidelines and Limitations	2-7
Default Settings	2-7

Configuring IPv4	2-7
Configuring IPv4 Addressing	2-8
Configuring Multiple IP Addresses	2-9
Configuring a Static ARP Entry	2-10
Configuring Proxy ARP	2-11
Configuring Local Proxy ARP	2-12
Configuring Gratuitous ARP	2-13
Configuring IP Directed Broadcasts	2-14
Configuring IP Glean Throttling	2-15
Configuring the Hardware IP Glean Throttle Maximum	2-16
Configuring a Hardware IP Glean Throttle Timeout	2-17
Configuring the Interface IP Address for the ICMP Source IP Field	2-18
Verifying the IPv4 Configuration	2-19
Configuration Examples for IPv4	2-19
Additional References	2-19
Related Documents	2-20
Standards	2-20

CHAPTER 3

Configuring IPv6 3-1

Information About IPv6	3-1
IPv6 Address Formats	3-2
IPv6 Unicast Addresses	3-3
Aggregatable Global Addresses	3-3
Link-Local Addresses	3-5
IPv4-Compatible IPv6 Addresses	3-5
Unique Local Addresses	3-6
Site-Local Address	3-7
IPv4 Packet Header	3-7
Simplified IPv6 Packet Header	3-7
DNS for IPv6	3-10
Path MTU Discovery for IPv6	3-10
CDP IPv6 Address Support	3-11
ICMP for IPv6	3-11
IPv6 Neighbor Discovery	3-12
IPv6 Neighbor Solicitation Message	3-12
IPv6 Router Advertisement Message	3-13
IPv6 Neighbor Redirect Message	3-15
Virtualization Support	3-16
IPv6 Routes with ECMP	3-16

Licensing Requirements for IPv6	3-16
Prerequisites for IPv6	3-16
Guidelines and Limitations for IPv6	3-16
Default Settings	3-17
Configuring IPv6	3-17
Configuring IPv6 Addressing	3-17
Configuring the System Routing Mode	3-19
Configuring ALPM	3-21
Configuring IPv6 Neighbor Discovery	3-23
Optional IPv6 Neighbor Discovery	3-25
Configuring IPv6 Packet Verification	3-26
Verifying the IPv6 Configuration	3-27
Configuration Examples for IPv6	3-27
Additional References	3-28
Related Documents	3-28
Standards	3-28
	3-28

CHAPTER 4

Configuring OSPFv2	4-1
Information About OSPFv2	4-1
Hello Packet	4-2
Neighbors	4-2
Adjacency	4-3
Designated Routers	4-3
Areas	4-4
Link-State Advertisements	4-5
LSA Types	4-5
Link Cost	4-6
Flooding and LSA Group Pacing	4-6
Link-State Database	4-7
Opaque LSAs	4-7
OSPFv2 and the Unicast RIB	4-7
Authentication	4-7
Simple Password Authentication	4-8
MD5 Authentication	4-8
Advanced Features	4-8
Stub Area	4-8
Not-So-Stubby Area	4-9
Virtual Links	4-9

Route Redistribution	4-10
Route Summarization	4-10
OSPFv2 Stub Router Advertisements	4-11
Multiple OSPFv2 Instances	4-11
SPF Optimization	4-11
BFD	4-11
Virtualization Support	4-12
Licensing Requirements for OSPFv2	4-12
Prerequisites for OSPFv2	4-12
Guidelines and Limitations	4-12
Default Settings	4-12
Configuring Basic OSPFv2	4-13
Enabling the OSPFv2 Feature	4-13
Creating an OSPFv2 Instance	4-14
Configuring Optional Parameters on an OSPFv2 Instance	4-15
Configuring Networks in OSPFv2	4-16
Configuring Authentication for an Area	4-18
Configuring Authentication for an Interface	4-20
Configuring Advanced OSPFv2	4-22
Configuring Filter Lists for Border Routers	4-23
Configuring Stub Areas	4-24
Configuring a Totally Stubby Area	4-25
Configuring NSSA	4-26
Configuring Virtual Links	4-28
Configuring Redistribution	4-30
Limiting the Number of Redistributed Routes	4-32
Configuring Route Summarization	4-34
Configuring Stub Route Advertisements	4-35
Modifying the Default Timers	4-36
Restarting an OSPFv2 Instance	4-39
Configuring OSPFv2 with Virtualization	4-39
Verifying the OSPFv2 Configuration	4-41
Displaying OSPFv2 Statistics	4-42
Configuration Examples for OSPFv2	4-42
Additional References	4-43
Related Documents	4-43
MIBs	4-43

CHAPTER 5**Configuring OSPFv3 5-1**

Information About OSPFv3	5-1
Comparison of OSPFv3 and OSPFv2	5-2
Hello Packet	5-2
Neighbors	5-3
Adjacency	5-3
Designated Routers	5-4
Areas	5-5
Link-State Advertisement	5-5
LSA Types	5-6
Link Cost	5-6
Flooding and LSA Group Pacing	5-7
Link-State Database	5-7
Multi-Area Adjacency	5-7
OSPFv3 and the IPv6 Unicast RIB	5-8
Address Family Support	5-8
Advanced Features	5-8
Stub Area	5-9
Not-So-Stubby Area	5-9
Virtual Links	5-10
Route Redistribution	5-10
Route Summarization	5-10
High Availability and Graceful Restart	5-11
Multiple OSPFv3 Instances	5-12
SPF Optimization	5-12
BFD	5-12
Virtualization Support	5-12
Licensing Requirements for OSPFv3	5-12
Prerequisites for OSPFv3	5-13
Guidelines and Limitations for OSPFv3	5-13
Default Settings	5-13
Configuring Basic OSPFv3	5-14
Enabling OSPFv3	5-14
Creating an OSPFv3 Instance	5-15
Configuring Networks in OSPFv3	5-17
Configuring Advanced OSPFv3	5-20
Configuring Filter Lists for Border Routers	5-20
Configuring Stub Areas	5-22
Configuring a Totally Stubby Area	5-23

Configuring NSSA	5-23
Configuring Multi-Area Adjacency	5-25
Configuring Virtual Links	5-27
Configuring Redistribution	5-29
Limiting the Number of Redistributed Routes	5-30
Configuring Route Summarization	5-32
Modifying the Default Timers	5-34
Configuring Graceful Restart	5-36
Restarting an OSPFv3 Instance	5-37
Configuring OSPFv3 with Virtualization	5-38
Verifying the OSPFv3 Configuration	5-40
Monitoring OSPFv3	5-41
Configuration Examples for OSPFv3	5-41
Related Topics	5-41
Additional References	5-41
Related Documents	5-42
MIBs	5-42

CHAPTER 6

Configuring EIGRP 6-1

Information About EIGRP	6-1
EIGRP Components	6-2
Reliable Transport Protocol	6-2
Neighbor Discovery and Recovery	6-2
Diffusing Update Algorithm	6-2
EIGRP Route Updates	6-3
Internal Route Metrics	6-3
External Route Metrics	6-4
EIGRP and the Unicast RIB	6-4
Advanced EIGRP	6-4
Address Families	6-4
Authentication	6-5
Stub Routers	6-5
Route Summarization	6-6
Route Redistribution	6-6
Load Balancing	6-6
Split Horizon	6-6
Virtualization Support	6-7
Licensing Requirements for EIGRP	6-7
Prerequisites for EIGRP	6-7

Guidelines and Limitations	6-7
Default Settings	6-8
Configuring Basic EIGRP	6-8
Enabling the EIGRP Feature	6-9
Creating an EIGRP Instance	6-10
Restarting an EIGRP Instance	6-12
Shutting Down an EIGRP Instance	6-12
Configuring a Passive Interface for EIGRP	6-12
Shutting Down EIGRP on an Interface	6-13
Configuring Advanced EIGRP	6-13
Configuring Authentication in EIGRP	6-13
Configuring EIGRP Stub Routing	6-16
Configuring a Summary Address for EIGRP	6-16
Redistributing Routes into EIGRP	6-17
Limiting the Number of Redistributed Routes	6-19
Configuring Load Balancing in EIGRP	6-21
Adjusting the Interval Between Hello Packets and the Hold Time	6-22
Disabling Split Horizon	6-23
Tuning EIGRP	6-23
Configuring Virtualization for EIGRP	6-25
Verifying the EIGRP Configuration	6-27
Displaying EIGRP Statistics	6-27
Configuration Examples for EIGRP	6-28
Related Topics	6-28
Additional References	6-28
Related Documents	6-28
MIBs	6-28

CHAPTER 7
Configuring Basic BGP 7-1

Information About Basic BGP	7-1
BGP Autonomous Systems	7-2
4-Byte AS Number Support	7-2
Administrative Distance	7-2
BGP Peers	7-3
BGP Sessions	7-3
Dynamic AS Numbers for Prefix Peers	7-3
BGP Router Identifier	7-3
BGP Path Selection	7-4

Step 1—Comparing Pairs of Paths	7-4
Step 2—Determining the Order of Comparisons	7-6
Step 3—Determining the Best-Path Change Suppression	7-6
BGP and the Unicast RIB	7-7
BGP Virtualization	7-7
Licensing Requirements for Basic BGP	7-7
Prerequisites for Basic BGP	7-7
Guidelines and Limitations for BGP	7-8
CLI Configuration Modes	7-8
Global Configuration Mode	7-8
Address Family Configuration Mode	7-9
Neighbor Configuration Mode	7-9
Neighbor Address Family Configuration Mode	7-9
Default Settings	7-10
Configuring Basic BGP	7-10
Enabling the BGP Feature	7-11
Creating a BGP Instance	7-12
Restarting a BGP Instance	7-13
Shutting Down BGP	7-14
Configuring BGP Peers	7-14
Distributing the Default Static Route to All BGP VRFs	7-16
Configuring Update Announcement Delay Timers	7-17
Configuring BGP Reconnect Interval	7-19
Configuring Dynamic AS Numbers for Prefix Peers	7-20
Clearing BGP Information	7-22
Verifying the Basic BGP Configuration	7-25
Displaying BGP Statistics	7-27
Configuration Examples for Basic BGP	7-27
Related Topics	7-27
Where to Go Next	7-27
Additional References	7-28
Related Documents	7-28
MIBs	7-28

CHAPTER 8**Configuring Advanced BGP 8-1**

Information About Advanced BGP	8-1
Peer Templates	8-2
Authentication	8-2

Route Policies and Resetting BGP Sessions	8-3
eBGP	8-3
eBGP Next-Hop Unchanged	8-4
iBGP	8-4
AS Confederations	8-5
Route Reflector	8-5
Capabilities Negotiation	8-6
Route Dampening	8-6
Load Sharing and Multipath	8-7
Route Aggregation	8-7
BGP Conditional Advertisement	8-7
BGP Next-Hop Address Tracking	8-8
Site of Origin	8-8
Route Redistribution	8-9
BFD	8-9
Tuning BGP	8-9
BGP Timers	8-9
Tuning the Best-Path Algorithm	8-10
Multiprotocol BGP	8-10
RFC 5549	8-10
Licensing Requirements for Advanced BGP	8-10
Prerequisites for Advanced BGP	8-10
Guidelines and Limitations for Advanced BGP	8-11
Default Settings	8-12
Configuring Advanced BGP	8-12
Enabling IP Forward on an Interface	8-13
Configuring BGP Session Templates	8-13
Configuring BGP Peer-Policy Templates	8-16
Configuring BGP Peer Templates	8-18
Configuring Prefix Peering	8-21
Configuring BGP Authentication	8-22
Resetting a BGP Session	8-22
Modifying the Next-Hop Address	8-23
Configuring BGP Next-Hop Address Tracking	8-23
Configuring Next-Hop Filtering	8-24
Shrinking Next-Hop Groups When A Session Goes Down	8-24
Disabling Capabilities Negotiation	8-25
Configuring eBGP	8-25
Configuring eBGP Next-Hop Unchanged	8-26

Disabling eBGP Single-Hop Checking	8-26
Configuring eBGP Multihop	8-27
Configuring eBGP Routes in the Same Autonomous System	8-27
Disabling a Fast External Failover	8-28
Limiting the AS-path Attribute	8-28
Configuring Local AS Support	8-28
Configuring AS Confederations	8-29
Configuring an Autonomous System Path Containing Your Own Autonomous System	8-30
Configuring Route Reflector	8-31
Configuring Route Dampening	8-33
Configuring Route Deletion Delay	8-33
Configuring Load Sharing and ECMP	8-34
Configuring Maximum Prefixes	8-34
Configuring Dynamic Capability	8-34
Configuring Aggregate Addresses	8-35
Suppressing BGP Routes	8-35
Configuring BGP Conditional Advertisement	8-36
Configuring Route Redistribution	8-38
Disabling BGP Dampening with Redistribution	8-39
Configuring Multiprotocol BGP	8-40
Configuring BGP Extended Community Site of Origin	8-42
Tuning BGP	8-42
Configuring Virtualization	8-46
Verifying the Advanced BGP Configuration	8-48
Displaying BGP Statistics	8-49
Related Topics	8-49
Additional References	8-50
Related Documents	8-50
MIBs	8-50

CHAPTER 9

Configuring BGP Additional Paths 9-1

Information About BGP Additional Paths	9-1
Overview	9-1
Path-Hiding Scenario	9-1
Suboptimal Hot-Potato Routing Scenario	9-2
Benefits of Additional BGP Paths	9-3
BGP Additional Paths Functionality	9-3
Additional Path Selection	9-4
Advertising a Subset of the Paths Selected	9-4

Guidelines and Limitations	9-4
Configuring BGP Additional Paths	9-4
Configuring BGP Additional Paths for each Address Family	9-5
Configuring BGP Additional Paths for each Neighbor	9-6
Configuring Additional Paths Using a Peer Policy Template	9-7
Filtering and Setting Actions for Additional Paths	9-9
Configuration Examples for BGP Additional Paths	9-10
BGP Additional Paths Send and Receive Capabilities	9-11
R1	9-11
R2	9-11
BGP Additional Paths Using a Peer Policy Template	9-11
Verifying the BGP Additional Paths Configuration	9-12

CHAPTER 10

Configuring ECMP Recovery	10-1
Information About ECMP Recovery	10-1
Guidelines and Limitations for ECMP Recovery	10-1
Default Settings	10-1
Configuring ECMP Recovery	10-2
Verifying ECMP Recovery Configuration	10-2
Configuration Examples for ECMP Recovery	10-3

CHAPTER 11

Configuring ECMP for Host Routes	11-1
Information About ECMP for Host Routes	11-1
Guidelines for ECMP for Host Routes	11-1
Licensing Requirements for ECMP for Host Routes	11-2
Prerequisites for ECMP for Host Routes	11-2
Default Settings	11-2
Configuring ECMP for Host Routes	11-2
Enabling the ECMP for Host Routes Feature	11-2
Disabling the EMCP for Host Routes Feature	11-3
Verifying the ECMP for Host Routes Configuration	11-4
Configuration Examples for ECMP for Host Routes	11-4
Additional References	11-5
Related Documents	11-5

CHAPTER 12

Configuring RIP	12-1
Information About RIP	12-1

RIP Overview	12-2
RIPv2 Authentication	12-2
Split Horizon	12-2
Route Filtering	12-3
Route Summarization	12-3
Route Redistribution	12-3
Load Balancing	12-3
Virtualization Support	12-4
Licensing Requirements for RIP	12-4
Prerequisites for RIP	12-4
Guidelines and Limitations	12-4
Default Settings	12-4
Configuring RIP	12-5
Enabling the RIP Feature	12-5
Creating a RIP Instance	12-6
Restarting a RIP Instance	12-8
Configuring RIP on an Interface	12-8
Configuring RIP Authentication	12-9
Configuring a Passive Interface	12-11
Configuring Split Horizon with Poison Reverse	12-11
Configuring Route Summarization	12-11
Configuring Route Redistribution	12-12
Configuring Virtualization	12-13
Tuning RIP	12-16
Verifying the RIP Configuration	12-17
Displaying RIP Statistics	12-17
Configuration Examples for RIP	12-18
Related Topics	12-18
Additional References	12-18
Related Documents	12-18
Standards	12-18

CHAPTER 13

Configuring Static Routing 13-1

Information About Static Routing	13-1
Administrative Distance	13-2
Directly Connected Static Routes	13-2
Fully Specified Static Routes	13-2
Floating Static Routes	13-2

Remote Next-hops for Static Routes	13-3
BFD	13-3
Virtualization Support	13-3
Licensing Requirements for Static Routing	13-3
Prerequisites for Static Routing	13-3
Guidelines and Limitations	13-3
Default Settings	13-4
Configuring Static Routing	13-4
Configuring a Static Route	13-4
Configuring Virtualization	13-5
Verifying the Static Routing Configuration	13-6
Configuration Examples for Static Routing	13-6
Additional References	13-7
Related Documents	13-7

CHAPTER 14

Configuring Layer 3 Virtualization	14-1
Layer 3 Virtualization	14-1
Overview of Layer 3 Virtualization	14-1
VRF and Routing	14-2
VRF-Lite	14-2
VRF-Aware Services	14-3
Reachability	14-3
Filtering	14-4
Combining Reachability and Filtering	14-4
Licensing Requirements for VRFs	14-5
Guidelines and Limitations for VRF	14-5
Guidelines and Limitations for VRF-Lite	14-5
Guidelines and Limitations for VRF Route Leaking	14-6
Default Settings	14-6
Configuring VRFs	14-6
Creating a VRF	14-7
Assigning VRF Membership to an Interface	14-8
Configuring VRF Parameters for a Routing Protocol	14-9
Configuring a VRF-Aware Service	14-11
Setting the VRF Scope	14-12
Configuring Non-recursive Static IP Route CLI in Default VRF	14-13
Configuring Recursive Static IP Route CLI in Default VRF	14-14
Configuring Non-recursive Static IP Route CLI in Management VRF	14-14

Configuring Recursive Static IP Route CLI in Management VRF	14-15
Configuring Non-recursive Static IPv6 Route CLI in Default VRFs	14-16
Configuring Recursive Static IPv6 Route CLI in Default VRF	14-16
Configuring Non-recursive Static IPv6 Route CLI in Management VRF	14-17
Configuring Recursive Static IPv6 Route CLI in Management VRF	14-18
Verifying the VRF Configuration	14-18
Configuration Examples for VRF	14-19
Additional References	14-21
Related Documents	14-21
Standards	14-22

CHAPTER 15
Configuring the Unicast RIB and FIB 15-1

Information About the Unicast RIB and FIB	15-1
Layer 3 Consistency Checker	15-2
FIB Tables	15-2
Virtualization Support	15-3
Licensing Requirements for the Unicast RIB and FIB	15-3
Configuring the Unicast RIB and FIB	15-3
Displaying Module FIB Information	15-3
Configuring Load Sharing in the Unicast FIB	15-4
Displaying Routing and Adjacency Information	15-7
Triggering the Layer 3 Consistency Checker	15-8
Clearing Forwarding Information in the FIB	15-9
Estimating Memory Requirements for Routes	15-9
Clearing Routes in the Unicast RIB	15-9
Verifying the Unicast RIB and FIB Configuration	15-10
Additional References	15-11
Related Documents	15-11

CHAPTER 16
Configuring Route Policy Manager 16-1

Information About Route Policy Manager	16-1
Prefix Lists	16-2
Prefix List Masks	16-2
MAC Lists	16-2
Route Maps	16-3
Match Criteria	16-3
Set Changes	16-3
Access Lists	16-4

AS Numbers for BGP	16-4
AS-Path Lists for BGP	16-4
Community Lists for BGP	16-4
Extended Community Lists for BGP	16-4
Route Redistribution and Route Maps	16-5
Licensing Requirements for Route Policy Manager	16-5
Guidelines and Limitations	16-5
Default Settings	16-6
Configuring Route Policy Manager	16-6
Configuring IP Prefix Lists	16-6
Configuring MAC Lists	16-8
Configuring AS-Path Lists	16-8
Configuring Community Lists	16-9
Configuring Extended Community Lists	16-11
Configuring Route Maps	16-12
Verifying the Route Policy Manager Configuration	16-17
Configuration Examples for Route Policy Manager	16-17
Related Topics	16-18
Additional References	16-18
Related Documents	16-18
Standards	16-18

CHAPTER 17

Configuring Policy-Based Routing	17-1
Information About Policy-Based Routing	17-1
Policy Route Maps	17-2
Set Criteria for Policy-Based Routing	17-2
Licensing Requirements for Policy-Based Routing	17-3
Prerequisites for Policy-Based Routing	17-3
Guidelines and Limitations for Policy-Based Routing	17-3
Default Settings	17-4
Configuring Policy-Based Routing	17-4
Enabling the Policy-Based Routing Feature	17-4
Configuring a Route Policy	17-5
Verifying the Policy-Based Routing Configuration	17-8
Displaying Policy-Based Routing Statistics	17-8
Clearing Policy-Based Routing Statistics	17-8
Configuration Examples for Policy Based-Routing	17-8

Related Topics	17-9
Additional References	17-9
Related Documents	17-10
Standards	17-10
Feature History for Policy-Based Routing	17-10

CHAPTER 18

Configuring Bidirectional Forwarding Detection 18-1

Information About BFD	18-1
Asynchronous Mode	18-1
BFD Detection of Failures	18-2
BFD Echo Function	18-2
Security	18-2
Virtualization Support	18-3
Licensing Requirements for BFD	18-3
Prerequisites for BFD	18-3
Guidelines and Limitations	18-3
Default Settings	18-4
Configuring BFD	18-5
Configuration Hierarchy	18-5
Task Flow for Configuring BFD	18-5
Enabling the BFD Feature	18-6
Configuring Global BFD Parameters	18-6
Configuring BFD on an Interface	18-8
Configuring BFD on a Port Channel	18-9
Configuring the BFD Echo Function	18-10
Configuring BFD on BGP	18-12
Configuring BFD on PIM	18-13
Configuring BFD on OSPFv2	18-14
Configuring BFD for Static Routes	18-15
Configuring BFD for IPv6	18-16
Configuring Global BFD Parameters for IPv6	18-16
Configuring Per Interface BFD Parameters for IPv6	18-17
Configuring BFD for IPv6 on OSPFv3	18-18
Configuring BFD on IPv6 Static Routes	18-19
Configuring BFD Echo Mode for IPv6	18-20
Configuring BFD Session Echo Interval for IPv6	18-21
Configuring a BFD Echo Interface for IPv6	18-21
Configuring BFD Slow Timer for IPv6	18-22

Verifying the BFD Configuration 18-23

Monitoring BFD 18-23

CHAPTER 20

Configuring HSRP 20-1

Information About HSRP 20-1

HSRP Overview 20-2

HSRP for IPv4 20-3

HSRP Versions 20-4

HSRP Authentication 20-4

HSRP Messages 20-4

HSRP Load Sharing 20-4

Object Tracking and HSRP 20-5

Virtualization Support 20-5

Licensing Requirements for HSRP 20-6

Prerequisites for HSRP 20-6

Guidelines and Limitations 20-6

Default Settings 20-6

Configuring HSRP 20-7

Enabling the HSRP Feature 20-7

Configuring the HSRP Version 20-8

Configuring an HSRP Group for IPv4 20-8

Configuring the HSRP Virtual MAC Address 20-10

Authenticating HSRP 20-10

Configuring HSRP Object Tracking 20-12

Configuring the HSRP Priority 20-14

Customizing HSRP 20-14

Verifying the HSRP Configuration 20-16

Configuration Examples for HSRP 20-16

Additional References 20-16

Related Documents 20-17

MIBs 20-17

CHAPTER 21

Configuring VRRP 21-1

Information About VRRP 21-1

VRRP Operation 21-2

VRRP Benefits 21-3

Multiple VRRP Groups 21-3

VRRP Router Priority and Preemption 21-4

VRRP Advertisements	21-5
VRRP Authentication	21-5
VRRPv3	21-5
Virtualization Support	21-5
Licensing Requirements for VRRP	21-5
Guidelines and Limitations	21-6
Default Settings	21-6
Configuring VRRP	21-7
Enabling the VRRP Feature	21-7
Configuring VRRP Groups	21-7
Configuring VRRP Priority	21-9
Configuring VRRP Authentication	21-10
Configuring Time Intervals for Advertisement Packets	21-12
Disabling Preemption	21-13
Configuring VRRP Interface State Tracking	21-15
Configuring VRRPv3	21-17
Enabling VRRPv3	21-17
Configuring VRRPv3 Groups	21-17
Configuring the Delay Period for FHRP Client Initialization	21-19
Configuring VRRPv3 Control Groups	21-20
Verifying the VRRPv2 Configuration	21-21
Verifying the VRRPv3 Configuration	21-22
Displaying VRRP Statistics	21-22
Configuration Examples for VRRPv2	21-22
Configuration Example for VRRPv3	21-24
Additional References	21-24
Related Documents	21-24

CHAPTER 22

Configuring Object Tracking 22-1

Information About Object Tracking	22-1
Object Tracking Overview	22-1
Object Track List	22-2
Virtualization Support	22-2
Licensing Requirements for Object Tracking	22-3
Guidelines and Limitations	22-3
Default Settings	22-3
Configuring Object Tracking	22-3
Configuring Object Tracking for an Interface	22-4

Configuring Object Tracking for Route Reachability	22-5
Configuring an Object Track List with a Boolean Expression	22-6
Configuring an Object Track List with a Percentage Threshold	22-7
Configuring an Object Track List with a Weight Threshold	22-8
Configuring an Object Tracking Delay	22-10
Configuring Object Tracking for a Nondefault VRF	22-12
Verifying the Object Tracking Configuration	22-13
Configuration Examples for Object Tracking	22-13
Related Topics	22-13
Additional References	22-13
Related Documents	22-14
Standards	22-14
BGP RFCs	1-1
First-Hop Redundancy Protocols RFCs	1-1
IP Services RFCs	1-2
OSPF RFCs	1-2
RIP RFCs	1-2



Preface

This document describes the configuration details for Cisco NX-OS Unicast in Cisco NX-OS switches. This chapter includes the following sections:

- Audience, page xxiii
- Document Conventions, page xxiii
- Preface, page xxiii

Audience

This publication is for experienced network administrators who configure and maintain Cisco Nexus Series switches.

Document Conventions

Command descriptions use these conventions:

Convention	Description
boldface font	Commands and keywords are in boldface.
<i>italic font</i>	Arguments for which you supply values are in italics.
[]	Elements in square brackets are optional.
[x y z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Screen examples use these conventions:

<code>screen font</code>	Terminal sessions and information that the switch displays are in screen font.
<code>boldface screen font</code>	Information that you must enter is in boldface screen font.
<i><code>italic screen font</code></i>	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.

[]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

This document uses the following conventions:



Note

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



Caution

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

Obtain Documentation and Submit a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see [What's New in Cisco Product Documentation](#).

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the [What's New in Cisco Product Documentation RSS feed](#). The RSS feeds are a free service.



New and Changed Information

This chapter provides release-specific information for new and changed feature in the *Cisco Nexus 3000 Series NX-OS Unicast Routing Configuration Guide, Release 6.x*. The latest version of this document is available at the following Cisco website:

http://www.cisco.com/en/US/products/ps11541/products_installation_and_configuration_guides_list.html

To check for additional information, see the *Cisco Nexus 3000 Series NX-OS Release Notes, Release 6.x* available at the following Cisco website:

http://www.cisco.com/en/US/products/ps11541/prod_release_notes_list.html

Table 1 provides an overview of the significant changes made to this configuration guide. The table does not provide an exhaustive list of all changes to this guide or all new features in a particular release.

Table 1 *New and Changed Features Information*

Feature	Description	Changed in Release	Where Documented
Load sharing option for ECMP based on the source IP address	Added support for a new load sharing option for ECMP, based on the source IP address.	6.0(2)U6(6)	Configuring Load Sharing in the Unicast FIB, page 4
Support for configuring static IP routes and static IPv6 routes in the default and management VRFs.	Added support for configuring static IP routes and static IPv6 routes in the default and management VRFs.	6.0(2)U6(4)	Chapter 14, “Configuring VRFs”
Support for BFD for PIM mode.	Added support for BFD for PIM mode.	6.0(2)U5(3)	Chapter 18, “Configuring BFD on PIM”
Support for ALPM Mode 4	Added support for ALPM Mode 4.	6.0(2)U5(1)	Chapter 3, “Configuring IPv6”
Remove private autonomous system (AS) numbers.	Added support for remove private autonomous system (AS) numbers.	6.0(2)U5(1)	Chapter 8, “Configuring Advanced BGP”
Support for new CLI default-information originate	A new CLI default-information originate is added to distribute the default static route from non-default VRF to all BGP (VRF) Virtual Router Context and to install that route in the local BGP route table for all the BGP VRFs.	6.0(2)U5(1)	Chapter 7, “Configuring Basic BGP”

Table 1 *New and Changed Features Information (continued)*

Feature	Description	Changed in Release	Where Documented
Hash offset and hash polynomial configuration	Added support for hash offset and hash polynomial configuration.	6.0(2)U5(1)	Chapter 15, “Configuring the Unicast RIB and FIB”
Configuring VRRPv3.	Added support for configuring VRRPv3.	6.0(2)U5(1)	Chapter 21, “Configuring VRRP”
BGP	Added the ability to configure the interval after which a BGP connection can reconnect.	6.0(2)U4(1)	Chapter 7, “Configuring Basic BGP”
BGP	Added the ability to delay route deletion from the hardware.	6.0(2)U4(1)	Chapter 8, “Configuring Advanced BGP”
BGP	Added the ability to configure BGP prefix peering wait timers.	6.0(2)U4(1)	Chapter 8, “Configuring Advanced BGP”
BGP	Added the ability to configure BGP to shrink ECMP groups in an accelerated way when a session goes down.	6.0(2)U4(1)	Chapter 8, “Configuring Advanced BGP”
BGP	Added the ability to configure a minimum route advertisement interval (RAI) between the sending of BGP routing updates.	6.0(2)U4(1)	Chapter 7, “Configuring Basic BGP”
BGP	Added the ability to advertise newly learned BGP routes (IPv4 and IPv6) only after these routes are confirmed by the Forwarding Information Base (FIB) and programmed in the hardware.	6.0(2)U4(1)	Chapter 8, “Configuring Advanced BGP”
BFD	Added support for IPv6 with BFD	6.0(2)U4(1)	Chapter 18, “Configuring Bidirectional Forwarding Detection,”
BGP	Added support for RFC 5549	6.0(2)U4(1)	Chapter 7, “Configuring Basic BGP,” Chapter 8, “Configuring Advanced BGP,”
Prefix Matching	Added support for masks for prefix lists.	6.0(2)U3(1)	Chapter 16, “Prefix Lists”
ECMP Recovery	Added the feature.	6.0(2)U3(1)	Chapter 10, “Configuring ECMP Recovery”
Unified Forwarding Table	Added Mode 0 support and configuration details.	6.0(2)U2(1)	Chapter 1, “Unified Forwarding Table”
Policy-Based Routing	Added the feature.	6.0(2)U2(1)	Chapter 17, “Configuring Policy-Based Routing”
Disabling BGP Dampening with Redistribution	Added information about disabling BGP dampening with redistribution and how to configure it.	6.0(2)U2(1)	Chapter 8, “Disabling BGP Dampening with Redistribution”
ICMP unreachable support to set source interface	Added information about ICMP unreachable support to set source interface, and how to configure the interface IP address for the ICMP source IP field.	6.0(2)U2(1)	Chapter 2, “Configuring IPv4”

Table 1 ***New and Changed Features Information (continued)***

Feature	Description	Changed in Release	Where Documented
eBGP next-hop unchanged	Added information about eBGP next-hop unchanged and how to configure it.	6.0(2)U2(1)	Chapter 8, “eBGP Next-Hop Unchanged”
BGP remove-private-as enhancements	Added guidelines for using the remove-private-as command.	6.0(2)U2(1)	Chapter 8, “Configuring Advanced BGP”
BGP Additional Paths	Added information on how to configure BGP Additional Paths.	6.0(2)U1(1)	Chapter 9, “Configuring BGP Additional Paths”
VRF Route Leaking	Added information on VRF-lite support for VRF route leaking. Included guidelines, limitations, and sample configurations.	6.0(2)U1(1)	Chapter 14, “Configuring Layer 3 Virtualization”





Overview

This chapter introduces the basic concepts for Layer 3 unicast routing protocols in Cisco NX-OS.

This chapter includes the following sections:

- [Information About Layer 3 Unicast Routing, page 1-1](#)
- [Routing Algorithms, page 1-8](#)
- [Layer 3 Virtualization, page 1-10](#)
- [Cisco NX-OS Forwarding Architecture, page 1-10](#)
- [Summary of Layer 3 Unicast Routing Features, page 1-13](#)
- [Related Topics, page 1-15](#)

Information About Layer 3 Unicast Routing

Layer 3 unicast routing involves two basic activities: determining optimal routing paths and packet switching. You can use routing algorithms to calculate the optimal path from the router to a destination. This calculation depends on the algorithm selected, route metrics, and other considerations such as load balancing and alternate path discovery.

This section includes the following topics:

- [Routing Fundamentals, page 1-2](#)
- [Packet Switching, page 1-2](#)
- [Routing Metrics, page 1-3](#)
- [Router IDs, page 1-5](#)
- [Autonomous Systems, page 1-5](#)
- [Convergence, page 1-6](#)
- [Load Balancing and Equal Cost Multipath, page 1-6](#)
- [Route Redistribution, page 1-6](#)
- [Administrative Distance, page 1-7](#)
- [Stub Routing, page 1-7](#)

Routing Fundamentals

Routing protocols use a *metric* to evaluate the best path to the destination. A metric is a standard of measurement, such as a path bandwidth, that routing algorithms use to determine the optimal path to a destination. To aid path determination, routing algorithms initialize and maintain routing tables, that contain route information such as the IP destination address and the address of the next router or *next-hop*. Destination and next-hop associations tell a router that an IP destination can be reached optimally by sending the packet to a particular router that represents the next-hop on the way to the final destination. When a router receives an incoming packet, it checks the destination address and attempts to associate this address with the next-hop. See the “[Unicast RIB](#)” section on page 1-10 for more information about the route table.

Routing tables can contain other information such as the data about the desirability of a path. Routers compare metrics to determine optimal routes, and these metrics differ depending on the design of the routing algorithm used. See the “[Routing Metrics](#)” section on page 1-3.

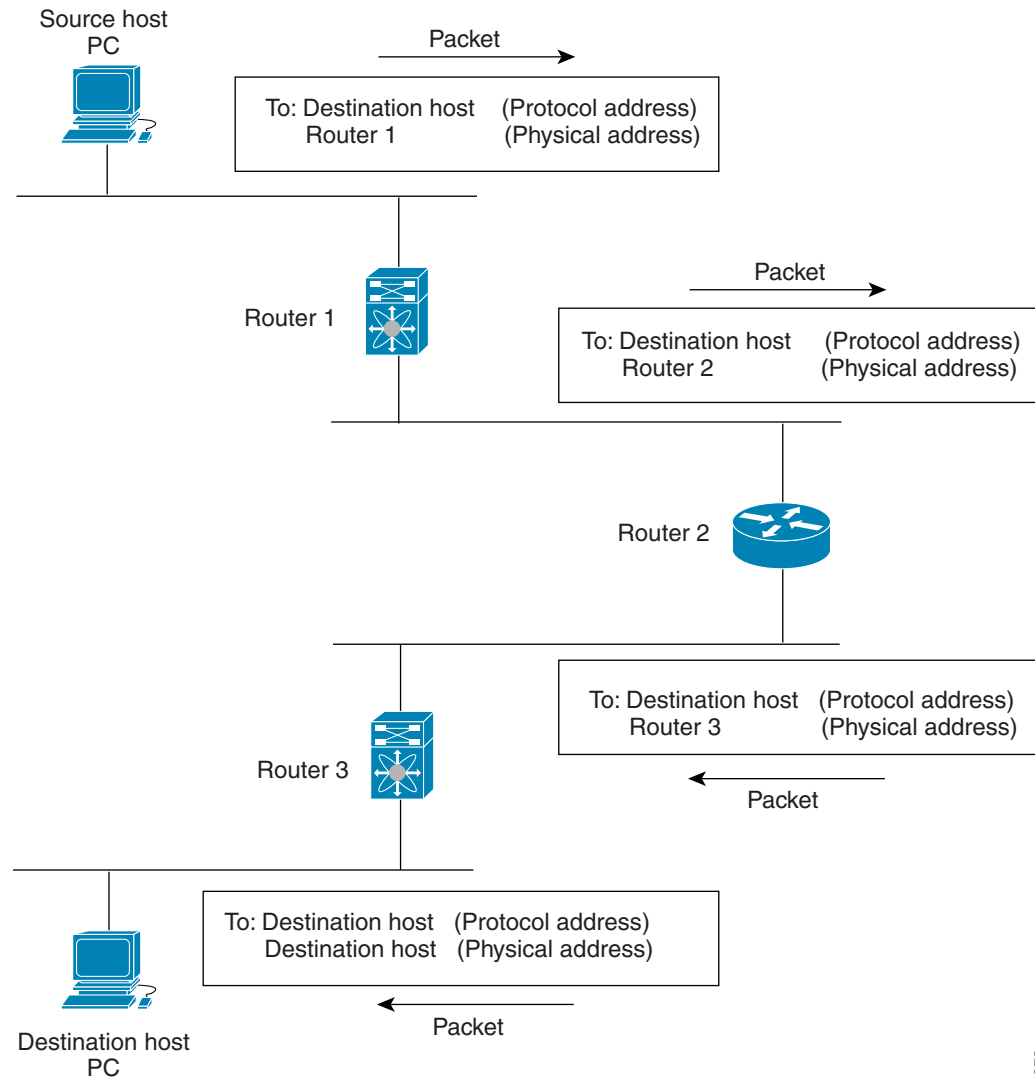
Routers communicate with one another and maintain their routing tables by transmitting a variety of messages. The routing update message is one of these messages that consists of all or a portion of a routing table. By analyzing routing updates from all other routers, a router can build a detailed picture of the network topology. A link-state advertisement, which is another example of a message sent between routers, informs other routers of the link state of the sending router. You can also use link information to enable routers to determine optimal routes to network destinations. For more information, see the “[Routing Algorithms](#)” section on page 1-8.

Packet Switching

In packet switching, a host determines that it must send a packet to another host. Having acquired a router address by some means, the source host sends a packet addressed specifically to the router physical (Media Access Control [MAC]-layer) address but with the IP (network layer) address of the destination host.

The router examines the destination IP address and tries to find the IP address in the routing table. If the router does not know how to forward the packet, it typically drops the packet. If the router knows how to forward the packet, it changes the destination MAC address to the MAC address of the next-hop router and transmits the packet.

The next-hop might be the ultimate destination host or another router that executes the same switching decision process. As the packet moves through the internetwork, its physical address changes, but its protocol address remains constant (see [Figure 1-1](#)).

Figure 1-1 *Packet Header Updates Through a Network*

182978

Routing Metrics

Routing algorithms use many different metrics to determine the best route. Sophisticated routing algorithms can base route selection on multiple metrics.

This section includes the following metrics:

- [Path Length, page 1-4](#)
- [Reliability, page 1-4](#)
- [Routing Delay, page 1-4](#)
- [Bandwidth, page 1-4](#)
- [Load, page 1-4](#)
- [Communication Cost, page 1-4](#)

Path Length

The *path length* is the most common routing metric. Some routing protocols allow you to assign arbitrary costs to each network link. In this case, the path length is the sum of the costs associated with each link traversed. Other routing protocols define hop count, a metric that specifies the number of passes through internetworking products, such as routers, that a packet must take from a source to a destination.

Reliability

The *reliability*, in the context of routing algorithms, is the dependability (in terms of the bit-error rate) of each network link. Some network links might go down more often than others. After a network fails, certain network links might be repaired more easily or more quickly than other links. The reliability factors that you can take into account when assigning the reliability rating are arbitrary numeric values that you usually assign to network links.

Routing Delay

The *routing delay* is the length of time required to move a packet from a source to a destination through the internetwork. The delay depends on many factors, including the bandwidth of intermediate network links, the port queues at each router along the way, the network congestion on all intermediate network links, and the physical distance that the packet needs to travel. Because the routing delay is a combination of several important variables, it is a common and useful metric.

Bandwidth

The *bandwidth* is the available traffic capacity of a link. For example, a 10-Gigabit Ethernet link would be preferable to a 1-Gigabit Ethernet link. Although the bandwidth is the maximum attainable throughput on a link, routes through links with greater bandwidth do not necessarily provide better routes than routes through slower links. For example, if a faster link is busier, the actual time required to send a packet to the destination could be greater.

Load

The *load* is the degree to which a network resource, such as a router, is busy. You can calculate the load in a variety of ways, including CPU utilization and packets processed per second. Monitoring these parameters on a continual basis can be resource intensive.

Communication Cost

The *communication cost* is a measure of the operating cost to route over a link. The communication cost is another important metric, especially if you do not care about performance as much as operating expenditures. For example, the line delay for a private line might be longer than a public line, but you can send packets over your private line rather than through the public lines that cost money for usage time.

Router IDs

Each routing process has an associated *router ID*. You can configure the router ID to any interface in the system. If you do not configure the router ID, Cisco NX-OS selects the router ID based on the following criteria:

- Cisco NX-OS prefers loopback0 over any other interface. If loopback0 does not exist, Cisco NX-OS prefers the first loopback interface over any other interface type.
- If you have not configured any loopback interfaces, Cisco NX-OS uses the first interface in the configuration file as the router ID. If you configure any loopback interface after Cisco NX-OS selects the router ID, the loopback interface becomes the router ID. If the loopback interface is not loopback0 and you configure loopback0 later with an IP address, the router ID changes to the IP address of loopback0.
- If the interface that the router ID is based on changes, that new IP address becomes the router ID. If any other interface changes its IP address, there is no router ID change.

Autonomous Systems

An *autonomous system* (AS) is a network controlled by a single technical administration entity. Autonomous systems divide global external networks into individual routing domains, where local routing policies are applied. This organization simplifies routing domain administration and simplifies consistent policy configuration.

Each autonomous system can support multiple interior routing protocols that dynamically exchange routing information through route *redistribution*. The Regional Internet Registries assign a unique number to each public autonomous system that directly connects to the Internet. This autonomous system number (AS number) identifies both the routing process and the autonomous system.

Cisco NX-OS supports 4-byte AS numbers. [Table 1-1](#) lists the AS number ranges.

Table 1-1 AS Numbers

2-Byte Numbers	4-Byte Numbers in AS.dot Notation	4-Byte Numbers in plaintext Notation	Purpose
1 to 64511	0.1 to 0.64511	1 to 64511	Public AS (assigned by RIR) ¹
64512 to 65534	0.64512 to 0.65534	64512 to 65534	Private AS (assigned by local administrator)
65535	0.65535	65535	Reserved
N/A	1.0 to 65535.65535	65536 to 4294967295	Public AS (assigned by RIR)

1. RIR=Regional Internet Registries

Private autonomous system numbers are used for internal routing domains but must be translated by the router for traffic that is routed out to the Internet. You should not configure routing protocols to advertise private autonomous system numbers to external networks. By default, Cisco NX-OS does not remove private autonomous system numbers from routing updates.

**Note**

The autonomous system number assignment for public and private networks is governed by the Internet Assigned Number Authority (IANA). For information about autonomous system numbers, including the reserved number assignment, or to apply to register an autonomous system number, refer to the following URL:

<http://www.iana.org/>

Convergence

A key aspect to measure for any routing algorithm is how much time a router takes to react to network topology changes. When a part of the network changes for any reason, such as a link failure, the routing information in different routers might not match. Some routers will have updated information about the changed topology, other routers will still have the old information. The *convergence* is the amount of time before all routers in the network have updated, matching routing information. The convergence time varies depending on the routing algorithm. Fast convergence minimizes the chance of lost packets caused by inaccurate routing information.

Load Balancing and Equal Cost Multipath

Routing protocols can use *load balancing* or equal cost multipath (ECMP) to share traffic across multiple paths. When a router learns multiple routes to a specific network, it installs the route with the lowest administrative distance in the routing table. If the router receives and installs multiple paths with the same administrative distance and cost to a destination, load balancing can occur. Load balancing distributes the traffic across all the paths, sharing the load. The number of paths used is limited by the number of entries that the routing protocol puts in the routing table. Cisco NX-OS supports up to 16 paths to a destination.

The Enhanced Interior Gateway Routing Protocol (EIGRP) also supports unequal cost load balancing. For more information, see [Chapter 6, “Configuring EIGRP.”](#)

Route Redistribution

If you have multiple routing protocols configured in your network, you can configure these protocols to share routing information by configuring route redistribution in each protocol. For example, you can configure Open Shortest Path First (OSPF) to advertise routes learned from the Border Gateway Protocol (BGP). You can also redistribute static routes into any dynamic routing protocol. The router that is redistributing routes from another protocol sets a fixed route metric for those redistributed routes and avoids the problem of incompatible route metrics between the different routing protocols. For example, routes redistributed from EIGRP into OSPF are assigned a fixed link cost metric that OSPF understands.

Route redistribution also uses an administrative distance (see the [“Administrative Distance” section on page 1-7](#)) to distinguish between routes learned from two different routing protocols. The preferred routing protocol is given a lower administrative distance so that its routes are chosen over routes from another protocol with a higher administrative distance assigned.

Administrative Distance

An *administrative distance* is a rating of the trustworthiness of a routing information source. A higher value means a lower trust rating. Typically, a route can be learned through more than one protocol. Administrative distance is used to discriminate between routes learned from more than one protocol. The route with the lowest administrative distance is installed in the IP routing table.

Stub Routing

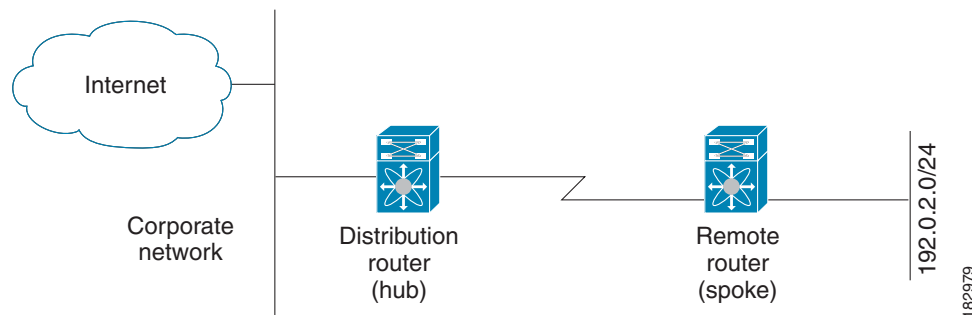
You can use stub routing in a hub-and-spoke network topology, where one or more end (stub) networks are connected to a remote router (the spoke) that is connected to one or more distribution routers (the hub). The remote router is adjacent only to one or more distribution routers. The only route for IP traffic to follow into the remote router is through a distribution router. This type of configuration is commonly used in WAN topologies in which the distribution router is directly connected to a WAN. The distribution router can be connected to many more remote routers. Often, the distribution router is connected to 100 or more remote routers. In a hub-and-spoke topology, the remote router must forward all nonlocal traffic to a distribution router, so it becomes unnecessary for the remote router to hold a complete routing table. Generally, the distribution router sends only a default route to the remote router.

Only specified routes are propagated from the remote (stub) router. The stub router responds to all queries for summaries, connected routes, redistributed static routes, external routes, and internal routes with the message “inaccessible.” A router that is configured as a stub sends a special peer information packet to all neighboring routers to report its status as a stub router.

Any neighbor that receives a packet informing it of the stub status does not query the stub router for any routes, and a router that has a stub peer does not query that peer. The stub router depends on the distribution router to send the proper updates to all peers.

Figure 1-2 shows a simple hub-and-spoke configuration.

Figure 1-2 Simple Hub-and-Spoke Network



Stub routing does not prevent routes from being advertised to the remote router. Figure 1-2 shows that the remote router can access the corporate network and the Internet through the distribution router only. A full route table on the remote router, in this example, serves no functional purpose because the path to the corporate network and the Internet would always be through the distribution router. A larger route table would reduce only the amount of memory required by the remote router. The bandwidth and memory used can be lessened by summarizing and filtering routes in the distribution router. In this network topology, the remote router does not need to receive routes that have been learned from other networks because the remote router must send all nonlocal traffic, regardless of its destination, to the distribution router. To configure a true stub network, you should configure the distribution router to send only a default route to the remote router.

OSPF supports stub areas and EIGRP supports stub routers.

Routing Algorithms

Routing algorithms determine how a router gathers and reports reachability information, how it deals with topology changes, and how it determines the optimal route to a destination. Various types of routing algorithms exist, and each algorithm has a different impact on network and router resources. Routing algorithms use a variety of metrics that affect calculation of optimal routes. You can classify routing algorithms by type, such as static or dynamic, and interior or exterior.

This section includes the following topics:

- [Static Routes and Dynamic Routing Protocols, page 1-8](#)
- [Interior and Exterior Gateway Protocols, page 1-8](#)
- [Distance Vector Protocols, page 1-9](#)
- [Link-State Protocols, page 1-9](#)

Static Routes and Dynamic Routing Protocols

Static routes are route table entries that you manually configure. These static routes do not change unless you reconfigure them. Static routes are simple to design and work well in environments where network traffic is relatively predictable and where network design is relatively simple.

Because static routing systems cannot react to network changes, you should not use them for today's large, constantly changing networks. Most routing protocols today use dynamic routing algorithms, which adjust to changing network circumstances by analyzing incoming routing update messages. If the message indicates that a network change has occurred, the routing software recalculates routes and sends out new routing update messages. These messages permeate the network, triggering routers to rerun their algorithms and change their routing tables accordingly.

You can supplement dynamic routing algorithms with static routes where appropriate. For example, you should configure each subnetwork with a static route to the IP *default gateway* or router of last resort (a router to which all unroutable packets are sent).

Interior and Exterior Gateway Protocols

You can separate networks into unique routing domains or autonomous systems. An autonomous system is a portion of an internetwork under common administrative authority that is regulated by a particular set of administrative guidelines. Routing protocols that route between autonomous systems are called exterior gateway protocols or interdomain protocols. BGP is an example of an exterior gateway protocol. Routing protocols used within an autonomous system are called interior gateway protocols or intradomain protocols. EIGRP and OSPF are examples of interior gateway protocols.

Distance Vector Protocols

Distance vector protocols use *distance vector* algorithms (also known as Bellman-Ford algorithms) that call for each router to send all or some portion of its routing table to its neighbors. Distance vector algorithms define routes by distance (for example, the number of hops to the destination) and direction (for example, the next-hop router). These routes are then broadcast to the directly connected neighbor routers. Each router uses these updates to verify and update the routing tables.

To prevent routing loops, most distance vector algorithms use *split horizon with poison reverse* which means that the routes learned from an interface are set as unreachable and advertised back along the interface that they were learned on during the next periodic update. This feature prevents the router from seeing its own route updates coming back.

Distance vector algorithms send updates at fixed intervals but can also send updates in response to changes in route metric values. These triggered updates can speed up the route convergence time. The Routing Information Protocol (RIP) is a distance vector protocol.

Link-State Protocols

The *link-state* protocols, also known as shortest path first (SPF), share information with neighboring routers. Each router builds a link-state advertisement (LSA), which contains information about each link and directly connected neighbor router.

Each LSA has a sequence number. When a router receives an LSA and updates its link-state database, the LSA is flooded to all adjacent neighbors. If a router receives two LSAs with the same sequence number (from the same router), the router does not flood the last LSA received to its neighbors to prevent an LSA update loop. Because the router floods the LSAs immediately after they receive them, convergence time for link-state protocols is minimized.

Discovering neighbors and establishing adjacency is an important part of a link state protocol. Neighbors are discovered using special Hello packets that also serve as keepalive notifications to each neighbor router. Adjacency is the establishment of a common set of operating parameters for the link-state protocol between neighbor routers.

The LSAs received by a router are added to its link-state database. Each entry consists of the following parameters:

- Router ID (for the router that originated the LSA)
- Neighbor ID
- Link cost
- Sequence number of the LSA
- Age of the LSA entry

The router runs the SPF algorithm on the link-state database, building the shortest path tree for that router. This SPF tree is used to populate the routing table.

In link-state algorithms, each router builds a picture of the entire network in its routing tables. The link-state algorithms send small updates everywhere, while distance vector algorithms send larger updates only to neighboring routers.

Because they converge more quickly, link-state algorithms are somewhat less prone to routing loops than distance vector algorithms. However, link-state algorithms require more CPU power and memory than distance vector algorithms. Link-state algorithms can be more expensive to implement and support. Link-state protocols are generally more scalable than distance vector protocols.

OSPF is an example of a link-state protocol.

Layer 3 Virtualization

Cisco NX-OS supports multiple Virtual Routing and Forwarding (VRF) instances and multiple routing information bases (*RIBs*) to support multiple address domains. Each VRF is associated with a RIB and this information is collected by the forwarding information base (FIB). A VRF represents a Layer 3 addressing domain. Each Layer 3 interface (logical or physical) belongs to one VRF. For more information, see [Chapter 14, “Configuring Layer 3 Virtualization.”](#)

Cisco NX-OS Forwarding Architecture

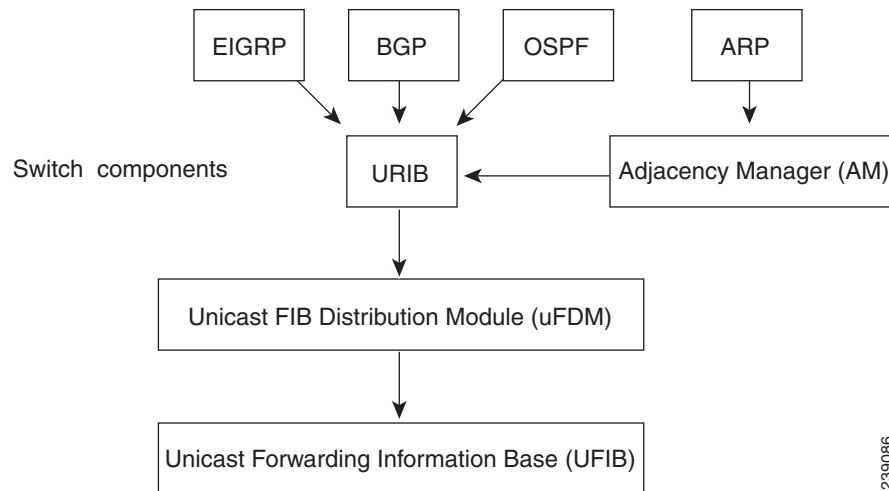
The Cisco NX-OS forwarding architecture is responsible for processing all routing updates and populating the forwarding information on the switch.

This section includes the following topics:

- [Unicast RIB, page 1-10](#)
- [Adjacency Manager, page 1-11](#)
- [Unicast Forwarding Distribution Module, page 1-11](#)
- [FIB, page 1-12](#)
- [Unified Forwarding Table, page 1-12](#)
- [Hardware Forwarding, page 1-12](#)
- [Software Forwarding, page 1-13](#)

Unicast RIB

The Cisco NX-OS forwarding architecture consists of multiple components, as shown in [Figure 1-3](#).

Figure 1-3 Cisco NX-OS Forwarding Architecture

The unicast RIB maintains the routing table with directly connected routes, static routes, and routes learned from dynamic unicast routing protocols. The unicast RIB also collects adjacency information from sources such as the Address Resolution Protocol (ARP). The unicast RIB determines the best next-hop for a given route and populates the unicast forwarding information base (FIB) by using the services of the unicast FIB distribution module (FDM).

Each dynamic routing protocol must update the unicast RIB for any route that has timed out. The unicast RIB then deletes that route and recalculates the best next-hop for that route (if an alternate path is available).

Adjacency Manager

The adjacency manager maintains adjacency information for different protocols including ARP, Open Shortest Path First version 2 (OSPFv2), Neighbor Discovery Protocol (NDP), and static configuration. The most basic adjacency information is the Layer 3 to Layer 2 address mapping discovered by these protocols. Outgoing Layer 2 packets use the adjacency information to complete the Layer 2 header.

The adjacency manager can trigger ARP requests to find a particular Layer 3 to Layer 2 mapping. The new mapping becomes available when the corresponding ARP reply is received and processed.

Unicast Forwarding Distribution Module

The unicast forwarding distribution module distributes the forwarding path information from the unicast RIB and other sources. The unicast RIB generates forwarding information which the unicast FIB programs into the hardware forwarding tables. The unicast forwarding distribution module also downloads the FIB information to newly inserted modules.

The unicast forwarding distribution module gathers adjacency information, rewrite information, and other platform-dependent information when updating routes in the unicast FIB. The adjacency and rewrite information consists of interface, next-hop, and Layer 3 to Layer 2 mapping information. The interface and next-hop information is received in route updates from the unicast RIB. The Layer 3 to Layer 2 mapping is received from the adjacency manager.

FIB

The unicast FIB builds the information used for the hardware forwarding engine. The unicast FIB receives route updates from the unicast forwarding distribution module and sends the information along to be programmed in the hardware forwarding engine. The unicast FIB controls the addition, deletion, and modification of routes, paths, and adjacencies.

The unicast FIBs are maintained on a per-VRF and per-address-family basis. Based on route update messages, the unicast FIB maintains a per-VRF prefix and next-hop adjacency information database. The next-hop adjacency data structure contains the next-hop IP address and the Layer 2 rewrite information. Multiple prefixes could share a next-hop adjacency information structure.

The unicast FIB also enables and disables unicast reverse path forwarding (RPF) checks per interface. The Cisco Nexus 3000 Series switch supports the following two RPF modes that can be configured on each ingress interface:

- **RPF Strict Check**—Packets that do not have a verifiable source address in the routers forwarding table or do not arrive on any of the return paths to the source are dropped.
- **RPF Loose Check**—Packets have a verifiable source address in the routers forwarding table and the source is reachable through a physical interface. The ingress interface that receives the packet does not need to match any of the interfaces in the FIB.

Unified Forwarding Table

Cisco Nexus 3100 Series switches support a minimum guaranteed number of entries for Layer 2, Layer 3 host, longest prefix matching (LPM) tables and a 256,000 x 105 bit shared table. You can assign shared table entries (at 64,000 entry granularity) for Layer 2 table, Layer 3 host table, and Layer 3 LPM table to extend the sizes of these tables based on the deployment.

The configuration that is currently supported for the mode 0 shared table entries for Cisco Nexus 3100 Series switches along with the configuration for Cisco Nexus 3000 Series switches is listed in the following table.

Table 1-2 Mode 0 Forwarding Table.

	L2 Table	L3 Host Table	LPM Table
Cisco Nexus 3000 Series	128,000	16,000	16,000 (IPv4), 8000 (IPv6 64 bit key)
Cisco Nexus 3100 Series	288,000	16,000	16,000 (IPv4), 8000 (IPv6 64 bit key)

Hardware Forwarding

Cisco NX-OS supports distributed packet forwarding. The ingress port takes relevant information from the packet header and passes the information to the local switching engine. The local switching engine does the Layer 3 lookup and uses this information to rewrite the packet header. The ingress module forwards the packet to the egress port. If the egress port is on a different module, the packet is forwarded using the switch fabric to the egress module. The egress module does not participate in the Layer 3 forwarding decision.

You also can use the **show platform fib** or **show platform forwarding** commands to display details on hardware forwarding.

Software Forwarding

The software forwarding path in Cisco NX-OS is used mainly to handle features that are not supported in the hardware or to handle errors encountered during the hardware processing. Typically, packets with IP options or packets that need fragmentation are passed to the CPU. The unicast RIB and the adjacency manager make the forwarding decisions based on the packets that should be switched in software or terminated.

Software forwarding is controlled by control plane policies and rate limiters.

Summary of Layer 3 Unicast Routing Features

This section provides a brief introduction to the Layer 3 unicast features and protocols supported in Cisco NX-OS.

This section includes the following topics:

- [OSPF, page 1-13](#)
- [EIGRP, page 1-13](#)
- [BGP, page 1-13](#)
- [RIP, page 1-14](#)
- [Static Routing, page 1-14](#)
- [Layer 3 Virtualization, page 1-14](#)
- [Route Policy Manager, page 1-14](#)

Layer 3 uses the IPv4 protocol. For more information, see [Chapter 2, “Configuring IPv4.”](#)

OSPF

The OSPF protocol is a link-state routing protocol used to exchange network reachability information within an autonomous system. Each OSPF router advertises information about its active links to its neighbor routers. Link information consists of the link type, the link metric, and the neighbor router connected to the link. The advertisements that contain this link information are called link-state advertisements. For more information, see [Chapter 4, “Configuring OSPFv2.”](#)

EIGRP

The EIGRP protocol is a unicast routing protocol that has the characteristics of both distance vector and link-state routing protocols. It is an improved version of IGRP, which is a Cisco proprietary routing protocol. EIGRP relies on its neighbors to provide the routes, typical to a distance vector routing protocol. It constructs the network topology from the routes advertised by its neighbors, similar to a link-state protocol, and uses this information to select loop-free paths to destinations. For more information, see [Chapter 6, “Configuring EIGRP.”](#)

BGP

The Border Gateway Protocol (BGP) is an inter-autonomous system routing protocol. A BGP router advertises network reachability information to other BGP routers using the Transmission Control Protocol (TCP) as its reliable transport mechanism. The network reachability information includes the

destination network prefix, a list of autonomous systems that needs to be traversed to reach the destination, and the next-hop router. Reachability information contains additional path attributes such as the preference to a route, origin of the route, community and others. For more information, see [Chapter 7, “Configuring Basic BGP”](#) and [Chapter 8, “Configuring Advanced BGP.”](#)

RIP

The Routing Information Protocol (RIP) is a distance-vector protocol that uses a hop count as its metric. RIP is widely used for routing traffic in the global Internet and is an Interior Gateway Protocol (IGP), which means that it performs routing within a single autonomous system. For more information, see [Chapter 12, “Configuring RIP.”](#)

Static Routing

Static routing allows you to enter a fixed route to a destination. This feature is useful for small networks where the topology is simple. Static routing is also used with other routing protocols to control default routes and route distribution. For more information, see [Chapter 13, “Configuring Static Routing.”](#)

Layer 3 Virtualization

Virtualization allows you to share physical resources across separate management domains.

Cisco NX-OS supports Layer 3 virtualization with VPN Routing and Forwarding (VRF). A VRF provides a separate address domain for configuring Layer 3 routing protocols. For more information, see [Chapter 14, “Configuring Layer 3 Virtualization.”](#)

Route Policy Manager

The Route Policy Manager provides a route filtering capability in Cisco NX-OS. It uses route maps to filter routes distributed across various routing protocols and between different entities within a given routing protocol. Filtering is based on specific match criteria, which is similar to packet filtering by access control lists. For more information, see [Chapter 16, “Configuring Route Policy Manager.”](#)

First-Hop Redundancy Protocols

A first-hop redundancy protocol (FHRP) allows you to provide redundant connections to your hosts. If an active first-hop router fails, the FHRP automatically selects a standby router to take over. You do not need to update the hosts with new IP addresses because the address is virtual and shared between each router in the FHRP group. For more information on the Hot Standby Router Protocol (HSRP), see [Chapter 20, “Configuring HSRP.”](#) For more information on the Virtual Router Redundancy Protocol (VRRP), see [Chapter 21, “Configuring VRRP.”](#)

Object Tracking

Object tracking allows you to track specific objects on the network, such as the interface line protocol state, IP routing, and route reachability, and take action when the tracked object's state changes. This feature allows you to increase the availability of the network and shorten recovery time if an object state goes down. For more information, see [Chapter 22, “Configuring Object Tracking.”](#)

Related Topics

The following Cisco documents are related to the Layer 3 features:

- Exploring Autonomous System Numbers:
http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_9-1/autonomous_system_numbers.html



PART 1

IP



Configuring IPv4

This chapter describes how to configure Internet Protocol version 4 (IPv4), which includes addressing, Address Resolution Protocol (ARP), and Internet Control Message Protocol (ICMP), on the Cisco NX-OS switch.

This chapter includes the following sections:

- [Information About IPv4, page 2-1](#)
- [Licensing Requirements for IPv4, page 2-7](#)
- [Prerequisites for IPv4, page 2-7](#)
- [Guidelines and Limitations, page 2-7](#)
- [Default Settings, page 2-7](#)
- [Configuring IPv4, page 2-7](#)
- [Verifying the IPv4 Configuration, page 2-19](#)
- [Configuration Examples for IPv4, page 2-19](#)
- [Additional References, page 2-19](#)

Information About IPv4

You can configure IP on the switch to assign IP addresses to network interfaces. When you assign IP addresses, you enable the interfaces and allow communication with the hosts on those interfaces.

You can configure an IP address as primary or secondary on a switch. An interface can have one primary IP address and multiple secondary addresses. All networking switches on an interface should share the same primary IP address because the packets that are generated by the switch always use the primary IPv4 address. Each IPv4 packet is based on the information from a source or destination IP address. See the [“Multiple IPv4 Addresses” section on page 2-2](#).

You can use a subnet to mask the IP addresses. A mask is used to determine what subnet an IP address belongs to. An IP address contains the network address and the host address. A mask identifies the bits that denote the network number in an IP address. When you use the mask to subnet a network, the mask is then referred to as a subnet mask. Subnet masks are 32-bit values that allow the recipient of IP packets to distinguish the network ID portion of the IP address from the host ID portion of the IP address.

The IP feature in the Cisco NX-OS system is responsible for handling IPv4 packets, as well as the forwarding of IPv4 packets, which includes IPv4 unicast route lookup, reverse path forwarding (RPF) checks, and software access control list (ACL) forwarding. The IP feature also manages the network interface IP address configuration, duplicate address checks, static routes, and packet send and receive interface for IP clients.

This section includes the following topics:

- [Multiple IPv4 Addresses, page 2-2](#)
- [Address Resolution Protocol, page 2-3](#)
- [ARP Caching, page 2-3](#)
- [Static and Dynamic Entries in the ARP Cache, page 2-3](#)
- [Devices That Do Not Use ARP, page 2-4](#)
- [Reverse ARP, page 2-4](#)
- [Proxy ARP, page 2-5](#)
- [Local Proxy ARP, page 2-5](#)
- [Gratuitous ARP, page 2-5](#)
- [Glean Throttling, page 2-5](#)
- [ICMP, page 2-6](#)
- [ICMP Unreachable Support to Set Source Interface, page 2-6](#)
- [Virtualization Support, page 2-6](#)
- [IPv4 Routes with ECMP, page 2-6](#)

Multiple IPv4 Addresses

The Cisco NX-OS system supports multiple IP addresses per interface. You can specify an unlimited number of secondary addresses for a variety of situations. The most common situations are as follows:

- When there are not enough host IP addresses for a particular network interface. For example, if your subnet allows up to 254 hosts per logical subnet, but on one physical subnet you must have 300 host addresses, then you can use secondary IP addresses on the routers or access servers to allow you to have two logical subnets using one physical subnet.
- Two subnets of a single network might otherwise be separated by another network. You can create a single network from subnets that are physically separated by another network by using a secondary address. In these instances, the first network is extended, or layered on top of the second network. A subnet cannot appear on more than one active interface of the router at a time.



Note

If any switch on a network segment uses a secondary IPv4 address, all other switches on that same network interface must also use a secondary address from the same network or subnet. The inconsistent use of secondary addresses on a network segment can quickly cause routing loops.

Address Resolution Protocol

Networking switches and Layer 3 switches use Address Resolution Protocol (ARP) to map IP (network layer) addresses to (Media Access Control [MAC]-layer) addresses to enable IP packets to be sent across networks. Before a switch sends a packet to another switch, it looks in its own ARP cache to see if there is a MAC address and corresponding IP address for the destination switch. If there is no entry, the source switch sends a broadcast message to every switch on the network.

Each switch compares the IP address to its own. Only the switch with the matching IP address replies to the switch that sends the data with a packet that contains the MAC address for the switch. The source switch adds the destination switch MAC address to its ARP table for future reference, creates a data-link header and trailer that encapsulates the packet, and proceeds to transfer the data. Figure 2-1 shows the ARP broadcast and response process.

Figure 2-1 ARP Process



When the destination switch lies on a remote network which is beyond another switch, the process is the same except that the switch that sends the data sends an ARP request for the MAC address of the default gateway. After the address is resolved and the default gateway receives the packet, the default gateway broadcasts the destination IP address over the networks connected to it. The switch on the destination switch network uses ARP to obtain the MAC address of the destination switch and delivers the packet. ARP is enabled by default.

The default system-defined CoPP policy rate-limits ARP broadcast packets. The default system-defined CoPP policy prevents an ARP broadcast storm from affecting the control plane traffic but does not affect bridged packets.

ARP Caching

ARP caching minimizes broadcasts and limits wasteful use of network resources. The mapping of IP addresses to MAC addresses occurs at each hop (switch) on the network for every packet sent over an internetwork, which may affect network performance.

ARP caching stores network addresses and the associated data-link addresses in memory for a period of time, which minimizes the use of valuable network resources to broadcast for the same address each time a packet is sent. You must maintain the cache entries since the cache entries are set to expire periodically because the information might become outdated. Every switch on a network updates its tables as addresses are broadcast.

Static and Dynamic Entries in the ARP Cache

You must manually configure the IP addresses, subnet masks, gateways, and corresponding MAC addresses for each interface of each switch when using static routes. Static routing enables more control but requires more work to maintain the route table. You must update the table each time you add or change routes.

Dynamic routing uses protocols that enable the switches in a network to exchange routing table information with each other. Dynamic routing is more efficient than static routing because the route table is automatically updated unless you add a time limit to the cache. The default time limit is 25 minutes but you can modify the time limit if the network has many routes that are added and deleted from the cache.

Devices That Do Not Use ARP

When a network is divided into two segments, a bridge joins the segments and filters traffic to each segment based on MAC addresses. The bridge builds its own address table that uses MAC addresses only, as opposed to a switch, which has an ARP cache that contains both IP addresses and the corresponding MAC addresses.

Passive hubs are central-connection switches that physically connect other switches in a network. They send messages out on all their ports to the switches and operate at Layer 1 but do not maintain an address table.

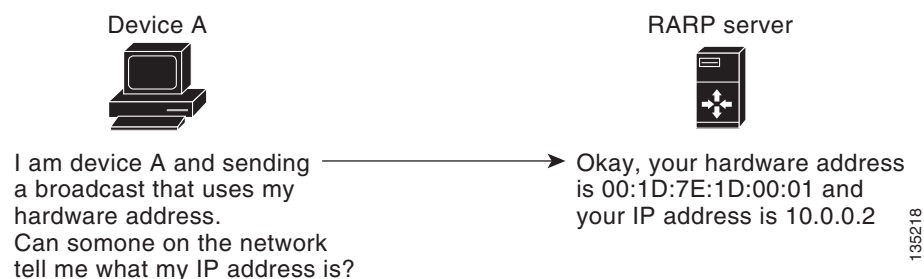
Layer 2 switches determine which port is connected to a device to which the message is addressed and send only to that port, unlike a hub, which sends the message out all of its ports. However, Layer 3 switches are switches that build an ARP cache (table).

Reverse ARP

Reverse ARP (RARP) as defined by RFC 903 works the same way as ARP, except that the RARP request packet requests an IP address instead of a MAC address. RARP often is used by diskless workstations because this type of device has no way to store IP addresses to use when they boot. The only address that is known is the MAC address because it is burned into the hardware.

Use of RARP requires an RARP server on the same network segment as the router interface. [Figure 2-2](#) illustrates how RARP works.

Figure 2-2 Reverse ARP



There are several limitations of RARP. Because of these limitations, most businesses use DHCP to assign IP addresses dynamically. DHCP is cost effective and requires less maintenance than RARP. The following are the most important limitations:

- Because RARP uses hardware addresses, if the internetwork is large with many physical networks, a RARP server must be on every segment with an additional server for redundancy. Maintaining two servers for every segment is costly.
- Each server must be configured with a table of static mappings between the hardware addresses and IP addresses. Maintenance of the IP addresses is difficult.

- RARP only provides IP addresses of the hosts and not subnet masks or default gateways.

Proxy ARP

Proxy ARP enables a switch that is physically located on one network appear to be logically part of a different physical network connected to the same switch or firewall. Proxy ARP allows you to hide a switch with a public IP address on a private network behind a router and still have the switch appear to be on the public network in front of the router. By hiding its identity, the router accepts responsibility for routing packets to the real destination. Proxy ARP can help switches on a subnet reach remote subnets without configuring routing or a default gateway.

When switches are not in the same data link layer network but in the same IP network, they try to transmit data to each other as if they are on the local network. However, the router that separates the switches does not send a broadcast message because routers do not pass hardware-layer broadcasts and the addresses cannot be resolved.

When you enable Proxy ARP on the switch and it receives an ARP request, it identifies the request as a request for a system that is not on the local LAN. The switch responds as if it is the remote destination for which the broadcast is addressed, with an ARP response that associates the MAC address of the switch with the IP address of the remote destination. The local switch believes that it is directly connected to the destination, while in reality its packets are being forwarded from the local subnetwork toward the destination subnetwork by their local switch. By default, Proxy ARP is disabled.

Local Proxy ARP

You can use local Proxy ARP to enable a switch to respond to ARP requests for IP addresses within a subnet where normally no routing is required. When you enable local Proxy ARP, ARP responds to all ARP requests for IP addresses within the subnet and forwards all traffic between hosts in the subnet. Use this feature only on subnets where hosts are intentionally prevented from communicating directly by the configuration on the switch to which they are connected.

Gratuitous ARP

Gratuitous ARP sends a request with identical source IP address and destination IP address to detect duplicate IP addresses. Cisco NX-OS supports enabling or disabling gratuitous ARP requests or ARP cache updates.

Glean Throttling

When forwarding an incoming IP packet, if the Address Resolution Protocol (ARP) request for the next-hop is not resolved, packets are punted to the central processing unit (CPU) for ARP resolution. The CPU resolves the MAC address for the next-hop and programs the hardware.

The Cisco Nexus 3000 Series device hardware has glean rate limiters to protect the supervisor from the glean traffic. If the maximum number of entries is exceeded, the packets for which the ARP request is not resolved continues to be processed in the software instead of getting dropped in the hardware.

When an ARP request is sent, the software adds a /32 drop adjacency in the hardware to prevent the packets to the same next-hop IP address to be forwarded to the supervisor. When the ARP is resolved, the hardware entry is updated with the correct MAC address. If the ARP entry is not resolved before a timeout period, the entry is removed from the hardware.

ICMP

You can use ICMP to provide message packets that report errors and other information that is relevant to IP processing. ICMP generates error messages, such as ICMP destination unreachable messages, ICMP Echo Requests (which send a packet on a round trip between two hosts) and Echo Reply messages. ICMP also provides many diagnostic functions and can send and redirect error packets to the host. By default, ICMP is enabled.

Some of the ICMP message types are as follows:

- Network error messages
- Network congestion messages
- Troubleshooting information
- Timeout announcements



Note

ICMP redirects are disabled on interfaces where the local proxy ARP feature is enabled.

ICMP Unreachable Support to Set Source Interface

You can configure an interface IP address for the ICMP source IP field to handle ICMP error messages. When ICMP packets are constructed in a network stack, the packets use the configured interface IP address. You can select Ethernet, loopback, or port channel interfaces to configure the IP address. If you do not select these options, the switch will behave as in earlier releases.

Virtualization Support

IPv4 supports Virtual Routing and Forwarding instances (VRFs). By default, Cisco NX-OS places you in the default VRF unless you specifically configure another VRF. For more information, see [Chapter 14, “Configuring Layer 3 Virtualization.”](#)

IPv4 Routes with ECMP

If all next-hops for a route are glean, drop, or punt, all next-hops are programmed as-is in the Multipath hardware table.

If some next-hops for a route are glean, drop, or punt, and the remaining next-hops are not, then only non glean, drop, or punt next-hops are programmed in the Multipath hardware table.

When a specific next-hop for ECMP route is resolved (ARP ND resolved), then the Multipath hardware table is updated accordingly.

Licensing Requirements for IPv4

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	IPv4 requires 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. For a complete explanation of the Cisco NX-OS licensing scheme, see the <i>Cisco NX-OS Licensing Guide</i> .

Prerequisites for IPv4

IPv4 has the following prerequisites:

- IPv4 can only be configured on Layer 3 interfaces.

Guidelines and Limitations

IPv4 has the following configuration guidelines and limitations:

- You can configure a secondary IP address only after you configure the primary IP address.
- If the Cisco Nexus 3000 switch is used as a Layer 2 or Layer 3 termination switch, Cisco recommends that you set the **mac-address-table-aging-time** to 1800 (higher than the default ARP aging time of 1500 seconds) on all VLANs.
- The Cisco Nexus 3000 switch does not support per-VLAN cam aging timers.

Default Settings

Table 2-1 lists the default settings for IP parameters.

Table 2-1 Default IP Parameters

Parameters	Default
ARP timeout	1500 seconds
proxy ARP	disabled

Configuring IPv4

This section includes the following topics:

- [Configuring IPv4 Addressing, page 2-8](#)
- [Configuring Multiple IP Addresses, page 2-9](#)
- [Configuring a Static ARP Entry, page 2-10](#)
- [Configuring Proxy ARP, page 2-11](#)

- [Configuring Local Proxy ARP, page 2-12](#)
- [Configuring Gratuitous ARP, page 2-13](#)
- [Configuring IP Directed Broadcasts, page 2-14](#)
- [Configuring IP Glean Throttling, page 2-15](#)
- [Configuring the Hardware IP Glean Throttle Maximum, page 2-16](#)
- [Configuring a Hardware IP Glean Throttle Timeout, page 2-17](#)

**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.

Configuring IPv4 Addressing

You can assign a primary IP address for a network interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *number*
3. **no switchport**
4. **ip address** *ip-address/length* [**secondary**]
5. (Optional) **show ip interface**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.

	Command	Purpose
Step 4	ip address <i>ip-address/length</i> [secondary] Example: switch(config-if)# ip address 192.2.1.1 255.0.0.0	Specifies a primary or secondary IPv4 address for an interface. <ul style="list-style-type: none"> The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means the corresponding address bit belongs to the network address. The network mask can be indicated as a slash (/) and a number - a prefix length. The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash must precede the decimal value and there is no space between the IP address and the slash.
Step 5	show ip interface Example: switch(config-if)# show ip interface	(Optional) Displays interfaces configured for IPv4.
Step 6	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to assign an IPv4 address:

```
switch# configure terminal
switch(config)# interface ethernet 2/3
switch(config-if)# no switchport
switch(config-if)# ip address 192.2.1.1 255.0.0.0
switch(config-if)# copy running-config startup-config
```

Configuring Multiple IP Addresses

You can only add secondary IP addresses after you configure primary IP addresses.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *number*
3. **no switchport**
4. **ip address** *ip-address/length* [**secondary**]
5. (Optional) **show ip interface**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	ip address <i>ip-address/length</i> [secondary] Example: switch(config-if)# ip address 192.2.1.1 255.0.0.0 secondary	Specifies the configured address as a secondary IPv4 address.
Step 5	show ip interface Example: switch(config-if)# show ip interface	(Optional) Displays interfaces configured for IPv4.
Step 6	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

Configuring a Static ARP Entry

You can configure a static ARP entry on the switch to map IP addresses to MAC hardware addresses, including static multicast MAC addresses.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *number*
3. **no switchport**
4. **ip arp** *ipaddr mac_addr*
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	ip arp <i>ipaddr mac_addr</i> Example: switch(config-if)# ip arp 192.2.1.1 0019.076c.1a78	Associates an IP address with a MAC address as a static entry.
Step 5	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure a static ARP entry:

```
switch# configure terminal
switch(config)# interface ethernet 2/3
switch(config-if)# no switchport
switch(config-if)# ip arp 192.2.1.1 0019.076c.1a78
switch(config-if)# copy running-config startup-config
```

Configuring Proxy ARP

You can configure Proxy ARP on the switch to determine the media addresses of hosts on other networks or subnets.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *number*
3. **no switchport**
4. **ip proxy-arp**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	ip proxy-arp Example: switch(config-if)# ip proxy-arp	Enables Proxy ARP on the interface.
Step 5	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure Proxy ARP:

```
switch# configure terminal
switch(config)# interface ethernet 2/3
switch(config-if)# no switchport
switch(config-if)# ip proxy-arp
switch(config-if)# copy running-config startup-config
```

Configuring Local Proxy ARP

You can configure Local Proxy ARP on the switch.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *number*
3. **no switchport**
4. **ip local-proxy-arp**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	ip local-proxy-arp Example: switch(config-if)# ip local-proxy-arp	Enables Local Proxy ARP on the interface.
Step 5	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure Local Proxy ARP:

```
switch# configure terminal
switch(config)# interface ethernet 2/3
switch(config-if)# no switchport
switch(config-if)# ip local-proxy-arp
switch(config-if)# copy running-config startup-config
```

Configuring Gratuitous ARP

You can configure gratuitous ARP on an interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *number*
3. **no switchport**
4. **ip arp gratuitous {request | update}**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	ip arp gratuitous { request update } Example: switch(config-if)# ip arp gratuitous request	Enables gratuitous ARP on the interface. Default is enabled.
Step 5	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to disable gratuitous ARP requests:

```
switch# configure terminal
switch(config)# interface ethernet 2/3
switch(config-if)# no switchport
switch(config-if)# no ip arp gratuitous request
switch(config-if)# copy running-config startup-config
```

Configuring IP Directed Broadcasts

An IP directed broadcast is an IP packet whose destination address is a valid broadcast address for some IP subnet, but which originates from a node that is not itself part of that destination subnet.

A switch that is not directly connected to its destination subnet forwards an IP directed broadcast in the same way it would forward unicast IP packets destined to a host on that subnet. When a directed broadcast packet reaches a switch that is directly connected to its destination subnet, that packet is “exploded” as a broadcast on the destination subnet. The destination address in the IP header of the packet is rewritten to the configured IP broadcast address for the subnet, and the packet is sent as a link-layer broadcast.

If directed broadcast is enabled for an interface, incoming IP packets whose addresses identify them as directed broadcasts intended for the subnet to which that interface is attached will be exploded as broadcasts on that subnet.

To enable IP directed broadcasts, use the following command in interface configuration mode:

Command	Purpose
<code>ip directed-broadcast</code>	Enables the translation of a directed broadcast to physical broadcasts

Configuring IP Glean Throttling

Cisco NX-OS software supports glean throttling rate limiters to protect the supervisor from the glean traffic.



Note

We recommend that you configure the IP glean throttle feature by using the **hardware ip glean throttle** command to filter the unnecessary glean packets that are sent to the supervisor for ARP resolution for the next-hops that are not reachable or do not exist. IP glean throttling boosts software performance and helps to manage traffic more efficiently.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **configure terminal**
2. **hardware ip glean throttle**
3. **no hardware ip glean throttle**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	<code>configure terminal</code> Example: <code>switch# configure terminal</code> <code>switch(config)#</code>	Enters global configuration mode.
Step 2	<code>hardware ip glean throttle</code> Example: <code>switch(config)# hardware ip glean throttle</code>	Enables ARP throttling.

	Command	Purpose
Step 3	no hardware ip glean throttle Example: switch(config)# no hardware ip glean throttle	Disables ARP throttling.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to enable IP glean throttling:

```
switch# configure terminal
switch(config)# hardware ip glean throttle
switch(config-if)# copy running-config startup-config
```

Configuring the Hardware IP Glean Throttle Maximum

You can limit the maximum number of drop adjacencies that are installed in the Forwarding Information Base (FIB).

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **configure terminal**
2. **hardware ip glean throttle maximum *count***
3. **no hardware ip glean throttle maximum *count***
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	hardware ip glean throttle maximum <i>count</i> Example: switch(config)# hardware ip glean throttle maximum 2134	Configures the number of drop adjacencies that are installed in the FIB.

	Command	Purpose
Step 3	no hardware ip glean throttle maximum count Example: switch(config)# no hardware ip glean throttle maximum 2134	Applies the default limits. The default value is 1000. The range is from 0 to 16,383 entries.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to limit the maximum number of drop adjacencies that are installed in the FIB:

```
switch# configure terminal
switch(config)# hardware ip glean throttle maximum 2134
switch(config-if)# copy running-config startup-config
```

Configuring a Hardware IP Glean Throttle Timeout

You can configure a timeout for the installed drop adjacencies to remain in the FIB.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **configure terminal**
2. **hardware ip glean throttle maximum timeout timeout-in-sec**
3. **no hardware ip glean throttle maximum timeout timeout-in-sec**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	hardware ip glean throttle maximum timeout timeout-in-sec Example: switch(config)# hardware ip glean throttle maximum timeout 300	Configures the timeout for the installed drop adjacencies to remain in the FIB.

	Command	Purpose
Step 3	no hardware ip glean throttle maximum timeout <i>timeout-in-sec</i> Example: switch(config)# no hardware ip glean throttle maximum timeout 300	Applies the default limits. The timeout value is in seconds. The range is from 300 seconds (5 minutes) to 1800 seconds (30 minutes). Note After the timeout period is exceeded, the drop adjacencies are removed from the FIB.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure a timeout for the drop adjacencies that are installed.

```
switch# configure terminal
switch(config)# hardware ip glean throttle maximum timeout 300
switch(config-if)# copy running-config startup-config
```

Configuring the Interface IP Address for the ICMP Source IP Field

You can configure an interface IP address for ICMP source IP field to handle ICMP error messages.

SUMMARY STEPS

1. configure terminal
2. [no] ip source {ethernet *slot/port* | loopback *number* | port-channel *number*} {icmp-errors}
3. exit

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[no] ip source {ethernet <i>slot/port</i> loopback <i>number</i> port-channel <i>number</i>} {icmp-errors} Example: switch(config)# ip source ethernet 1/1 icmp-errors	Configures an interface IP address for ICMP source IP field to route the ICMP error messages.
Step 3	exit Example: switch(config)# exit	Returns to privileged EXEC mode.

This example shows how to configure an interface IP address for the ICMP source IP field:

```
switch# configure terminal
switch(config)# ip source ethernet 1/1 icmp-errors
switch(config-if)# exit
```

This example shows how to unprovision an interface IP address from the ICMP source IP field:

```
switch# configure terminal
switch(config)# no ip source ethernet 1/1 icmp-errors
switch(config-if)# exit
```

Verifying the IPv4 Configuration

To display the IPv4 configuration, perform one of the following tasks:

Command	Purpose
show hardware forwarding ip verify	Displays the IP packet verification configuration.
show ip adjacency	Displays the adjacency table.
show ip arp	Displays the ARP table.
show ip interface	Displays IP-related interface information.
show ip arp statistics [vrf vrf-name]	Displays the ARP statistics.
show ip adjacency summary	Displays the summary of number of throttle adjacencies.
show ip arp summary	Displays the summary of the number of throttle adjacencies.
show ip adjacency throttle statistics	Displays only the throttled adjacencies.

Configuration Examples for IPv4

This example shows how to configure an IPv4 address:

```
configure terminal
interface ethernet 1/2
no switchport
ip address 192.2.1.1/16
```

Additional References

For additional information related to implementing IP, see the following sections:

- [Related Documents, page 2-20](#)
- [Standards, page 2-20](#)

Related Documents

Related Topic	Document Title
IP CLI commands	<i>Cisco Nexus 3000 Series Command Reference</i>

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—



Configuring IPv6

This chapter describes how to configure Internet Protocol version 6 (IPv6), which includes addressing, Neighbor Discovery Protocol (ND), and Internet Control Message Protocol version 6 (ICMPv6), on the Cisco NX-OS device.

This chapter includes the following sections:

- [Information About IPv6, page 3-1](#)
- [Licensing Requirements for IPv6, page 3-16](#)
- [Prerequisites for IPv6, page 3-16](#)
- [Guidelines and Limitations for IPv6, page 3-16](#)
- [Default Settings, page 3-17](#)
- [Configuring IPv6, page 3-17](#)
- [Verifying the IPv6 Configuration, page 3-27](#)
- [Configuration Examples for IPv6, page 3-27](#)
- [Additional References, page 3-28](#)

Information About IPv6

IPv6, which is designed to replace IPv4, increases the number of network address bits from 32 bits (in IPv4) to 128 bits. IPv6 is based on IPv4 but it includes a much larger address space and other improvements such as a simplified main header and extension headers.

The larger IPv6 address space allows networks to scale and provide global reachability. The simplified IPv6 packet header format handles packets more efficiently. The flexibility of the IPv6 address space reduces the need for private addresses and the use of Network Address Translation (NAT), which translates private (not globally unique) addresses into a limited number of public addresses. IPv6 enables new application protocols that do not require special processing by border routers at the edge of networks.

IPv6 functionality, such as prefix aggregation, simplified network renumbering, and IPv6 site multihoming capabilities, enable more efficient routing. IPv6 supports Open Shortest Path First (OSPF) for IPv6 and multiprotocol Border Gateway Protocol (BGP).

This section includes the following topics:

- [IPv6 Address Formats, page 3-2](#)

- [IPv6 Unicast Addresses, page 3-3](#)
- [IPv4 Packet Header, page 3-7](#)
- [IPv4 Packet Header, page 3-7](#)
- [Simplified IPv6 Packet Header, page 3-7](#)
- [DNS for IPv6, page 3-10](#)
- [Path MTU Discovery for IPv6, page 3-10](#)
- [CDP IPv6 Address Support, page 3-11](#)
- [ICMP for IPv6, page 3-11](#)
- [IPv6 Neighbor Discovery, page 3-12](#)
- [IPv6 Neighbor Solicitation Message, page 3-12](#)
- [IPv6 Router Advertisement Message, page 3-13](#)
- [IPv6 Neighbor Redirect Message, page 3-15](#)
- [Virtualization Support, page 3-16](#)
- [IPv6 Routes with ECMP, page 3-16](#)

IPv6 Address Formats

An IPv6 address has 128 bits or 16 bytes. The address is divided into eight, 16-bit hexadecimal blocks separated by colons (:) in the format: x:x:x:x:x:x:x:x. Two examples of IPv6 addresses are as follows:

```
2001:0DB8:7654:3210:FEDC:BA98:7654:3210
2001:0DB8:0:0:8:800:200C:417A
```

IPv6 addresses contain consecutive zeros within the address. You can use two colons (::) at the beginning, middle, or end of an IPv6 address to replace the consecutive zeros. [Table 3-1](#) shows a list of compressed IPv6 address formats.



Note

You can use two colons (::) only once in an IPv6 address to replace the longest string of consecutive zeros within the address.

You can use a double colon as part of the IPv6 address when consecutive 16-bit values are denoted as zero. You can configure multiple IPv6 addresses per interface but only one link-local address.

The hexadecimal letters in IPv6 addresses are not case sensitive.

Table 3-1 Compressed IPv6 Address Formats

IPv6 Address Type	Preferred Format	Compressed Format
Unicast	2001:0:0:0:0DB8:800:200C:417A	2001::0DB8:800:200C:417A
Multicast	FF01:0:0:0:0:0:0:101	FF01::101
Loopback	0:0:0:0:0:0:0:1	::1
Unspecified	0:0:0:0:0:0:0:0	::

A node may use the loopback address listed in [Table 3-1](#) to send an IPv6 packet to itself. The loopback address in IPv6 is the same as the loopback address in IPv4. For more information, see [Chapter 1, “Overview.”](#)

**Note**

You cannot assign the IPv6 loopback address to a physical interface. A packet that contains the IPv6 loopback address as its source or destination address must remain within the node that created the packet. IPv6 routers do not forward packets that have the IPv6 loopback address as their source or destination address.

**Note**

You cannot assign an IPv6 unspecified address to an interface. You should not use the unspecified IPv6 addresses as destination addresses in IPv6 packets or the IPv6 routing header.

The IPv6-prefix is in the form documented in RFC 2373 where the IPv6 address is specified in hexadecimal using 16-bit values between colons. The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). For example, 2001:0DB8:8086:6502::/32 is a valid IPv6 prefix.

IPv6 Unicast Addresses

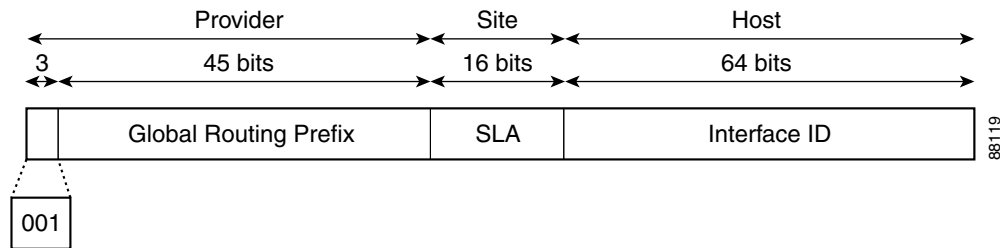
An IPv6 unicast address is an identifier for a single interface on a single node. A packet that is sent to a unicast address is delivered to the interface identified by that address. This section includes the following topics:

- [Aggregatable Global Addresses, page 3-3](#)
- [Link-Local Addresses, page 3-5](#)
- [IPv4-Compatible IPv6 Addresses, page 3-5](#)
- [Unique Local Addresses, page 3-6](#)
- [Site-Local Address, page 3-7](#)

Aggregatable Global Addresses

An aggregatable global address is an IPv6 address from the aggregatable global unicast prefix. The structure of aggregatable global unicast addresses enables strict aggregation of routing prefixes that limits the number of routing table entries in the global routing table. Aggregatable global addresses are used on links that are aggregated upward through organizations and eventually to the Internet service providers (ISPs).

Aggregatable global IPv6 addresses are defined by a global routing prefix, a subnet ID, and an interface ID. Except for addresses that start with binary 000, all global unicast addresses have a 64-bit interface ID. The IPv6 global unicast address allocation uses the range of addresses that start with binary value 001 (2000::/3). [Figure 3-1](#) shows the structure of an aggregatable global address.

Figure 3-1 **Aggregatable Global Address Format**

Addresses with a prefix of 2000::/3 (001) through E000::/3 (111) are required to have 64-bit interface identifiers in the extended universal identifier (EUI)-64 format. The Internet Assigned Numbers Authority (IANA) allocates the IPv6 address space in the range of 2000::/16 to regional registries.

The aggregatable global address consists of a 48-bit global routing prefix and a 16-bit subnet ID or Site-Level Aggregator (SLA). In the IPv6 aggregatable global unicast address format document (RFC 2374), the global routing prefix included two other hierarchically structured fields called Top-Level Aggregator (TLA) and Next-Level Aggregator (NLA). The IETF decided to remove the TLA and NLA fields from the RFCs because these fields are policy based. Some existing IPv6 networks deployed before the change might still use networks that are on the older architecture.

A subnet ID, which is a 16-bit subnet field, can be used by individual organizations to create a local addressing hierarchy and to identify subnets. A subnet ID is similar to a subnet in IPv4, except that an organization with an IPv6 subnet ID can support up to 65,535 individual subnets.

An interface ID identifies interfaces on a link. The interface ID is unique to the link. In many cases, an interface ID is the same as or based on the link-layer address of an interface. Interface IDs used in aggregatable global unicast and other IPv6 address types have 64 bits and are in the modified EUI-64 format.

Interface IDs are in the modified EUI-64 format in one of the following ways:

- For all IEEE 802 interface types (for example, Ethernet, and Fiber Distributed Data interfaces), the first three octets (24 bits) are the Organizationally Unique Identifier (OUI) of the 48-bit link-layer address (MAC address) of the interface, the fourth and fifth octets (16 bits) are a fixed hexadecimal value of FFFE, and the last three octets (24 bits) are the last three octets of the MAC address. The Universal/Local (U/L) bit, which is the seventh bit of the first octet, has a value of 0 or 1. Zero indicates a locally administered identifier; 1 indicates a globally unique IPv6 interface identifier.
- For all other interface types (for example, serial, loopback, ATM, Frame Relay, and tunnel interface types—except tunnel interfaces used with IPv6 overlay tunnels), the interface ID is similar to the interface ID for IEEE 802 interface types; however, the first MAC address from the pool of MAC addresses in the router is used as the identifier (because the interface does not have a MAC address).
- For tunnel interface types that are used with IPv6 overlay tunnels, the interface ID is the IPv4 address assigned to the tunnel interface with all zeros in the high-order 32 bits of the identifier.

**Note**

For interfaces that use the Point-to-Point Protocol (PPP), where the interfaces at both ends of the connection might have the same MAC address, the interface identifiers at both ends of the connection are negotiated (picked randomly and, if necessary, reconstructed) until both identifiers are unique. The first MAC address in the router is used as the identifier for interfaces using PPP.

If no IEEE 802 interface types are in the router, link-local IPv6 addresses are generated on the interfaces in the router in the following sequence:

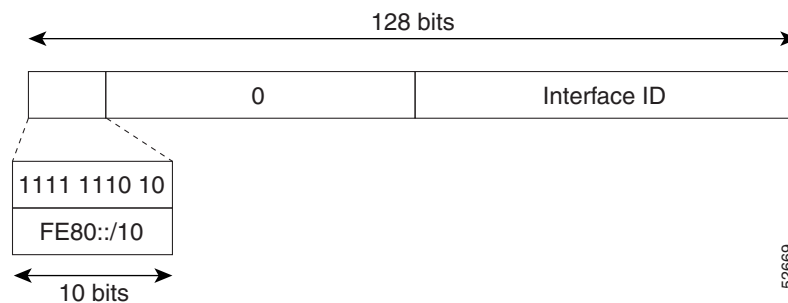
1. The router is queried for MAC addresses (from the pool of MAC addresses in the router).
2. If no MAC addresses are available in the router, the serial number of the router is used to form the link-local addresses.
3. If the serial number of the router cannot be used to form the link-local addresses, the router uses a Message Digest 5 (MD5) hash to determine the MAC address of the router from the hostname of the router.

Link-Local Addresses

A link-local address is an IPv6 unicast address that can be automatically configured on any interface using the link-local prefix FE80::/10 (1111 1110 10) and the interface identifier in the modified EUI-64 format. Link-local addresses are used in the Neighbor Discovery Protocol (NDP) and the stateless autoconfiguration process. Nodes on a local link can use link-local addresses to communicate; the nodes do not need globally unique addresses to communicate. [Figure 3-2](#) shows the structure of a link-local address.

IPv6 routers cannot forward packets that have link-local source or destination addresses to other links.

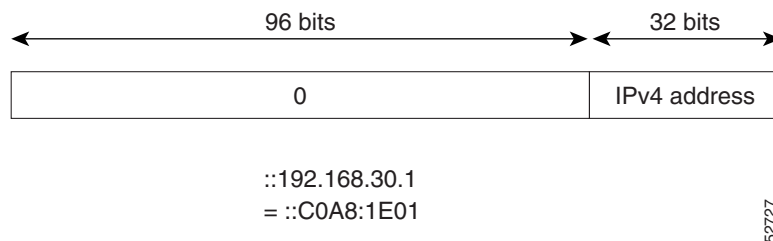
Figure 3-2 Link-Local Address Format



52669

IPv4-Compatible IPv6 Addresses

An IPv4-compatible IPv6 address is an IPv6 unicast address that has zeros in the high-order 96 bits of the address and an IPv4 address in the low-order 32 bits of the address. The format of an IPv4-compatible IPv6 address is 0:0:0:0:0:A.B.C.D or ::A.B.C.D. The entire 128-bit IPv4-compatible IPv6 address is used as the IPv6 address of a node and the IPv4 address embedded in the low-order 32 bits is used as the IPv4 address of the node. IPv4-compatible IPv6 addresses are assigned to nodes that support both the IPv4 and IPv6 protocol stacks and are used in automatic tunnels. [Figure 3-3](#) shows the structure of an IPv4-compatible IPv6 address and a few acceptable formats for the address.

Figure 3-3 IPv4-Compatible IPv6 Address Format

52727

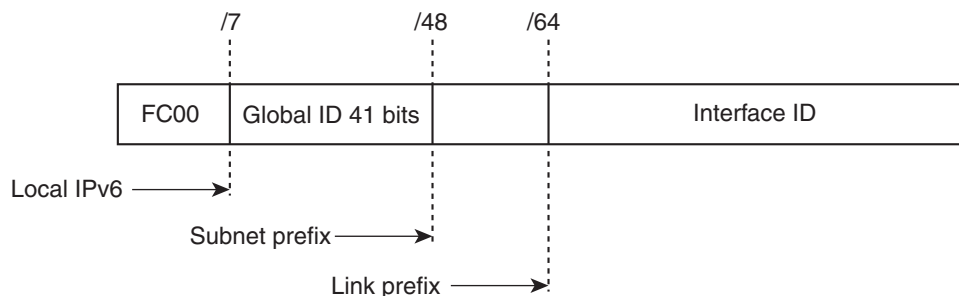
Unique Local Addresses

A unique local address is an IPv6 unicast address that is globally unique and is intended for local communications. It is not expected to be routable on the global Internet and is routable inside of a limited area, such as a site, and it may be routed between a limited set of sites. Applications may treat unique local addresses like global scoped addresses.

A unique local address has the following characteristics:

- It has a globally unique prefix (it has a high probability of uniqueness).
- It has a well-known prefix to allow for easy filtering at site boundaries.
- It allows sites to be combined or privately interconnected without creating any address conflicts or requiring renumbering of interfaces that use these prefixes.
- It is ISP-independent and can be used for communications inside of a site without having any permanent or intermittent Internet connectivity.
- If it is accidentally leaked outside of a site through routing or the Domain Name Server (DNS), there is no conflict with any other addresses.

Figure 3-4 shows the structure of a unique local address.

Figure 3-4 Unique Local Address Structure

- Prefix — FC00::/7 prefix to identify local IPv6 unicast addresses.
- Global ID — 41-bit global identifier used to create a globally unique prefix.
- Subnet ID — 16-bit subnet ID is an identifier of a subnet within the site.
- Interface ID — 64-bit ID

232389

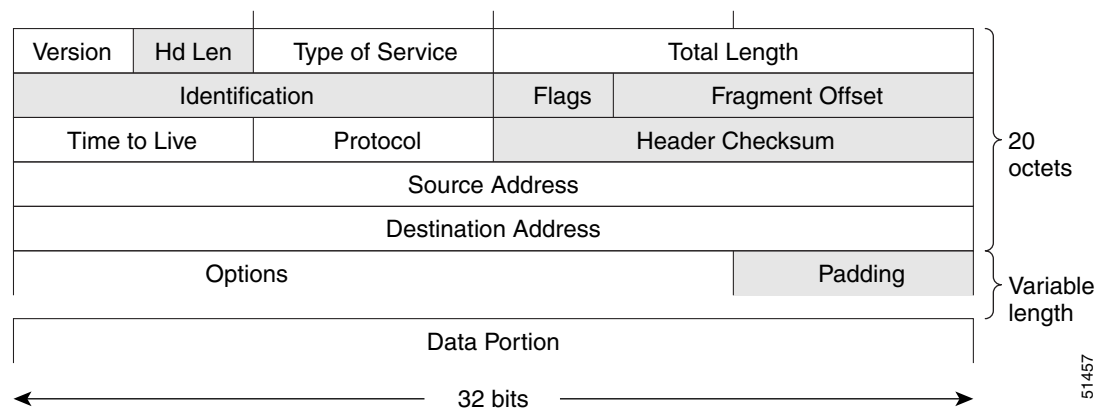
Site-Local Address

Because RFC 3879 deprecates the use of site-local addresses, you should follow the recommendations of unique local addressing (ULA) in RFC 4193 when you configure private IPv6 addresses.

IPv4 Packet Header

The base IPv4 packet header has 12 fields with a total size of 20 octets (160 bits) (see [Figure 3-5](#)). The 12 fields may be followed by an Options field, which is followed by a data portion that is usually the transport-layer packet. The variable length of the Options field adds to the total size of the IPv4 packet header. The shaded fields of the IPv4 packet header are not included in the IPv6 packet header.

Figure 3-5 IPv4 Packet Header Format



Simplified IPv6 Packet Header

The base IPv6 packet header has 8 fields with a total size of 40 octets (320 bits) (see [Figure 3-6](#)). Fragmentation is handled by the source of a packet and checksums at the data link layer and transport layer are used. The User Datagram Protocol (UDP) checksum checks the integrity of the inner packet and the base IPv6 packet header and Options field are aligned to 64 bits, which can facilitate the processing of IPv6 packets.

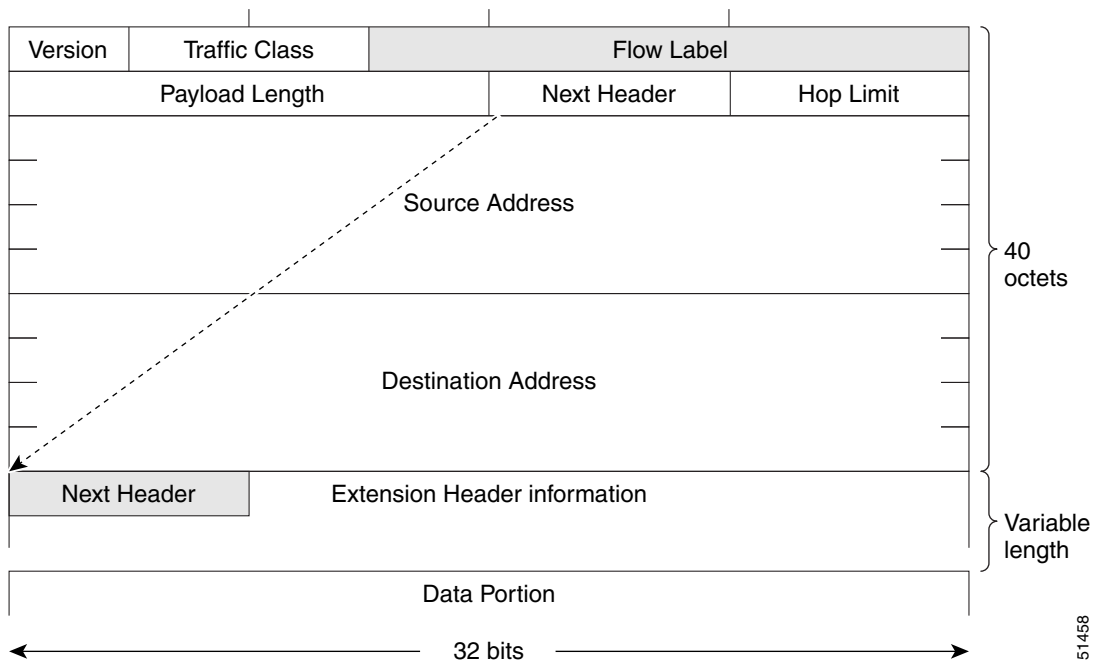
[Table 3-2](#) lists the fields in the base IPv6 packet header.

Table 3-2 Base IPv6 Packet Header Fields

Field	Description
Version	Similar to the Version field in the IPv4 packet header, except that the field lists number 6 for IPv6 instead of number 4 for IPv4.
Traffic Class	Similar to the Type of Service field in the IPv4 packet header. The Traffic Class field tags packets with a traffic class that is used in differentiated services.
Flow Label	New field in the IPv6 packet header. The Flow Label field tags packets with a specific flow that differentiates the packets at the network layer.

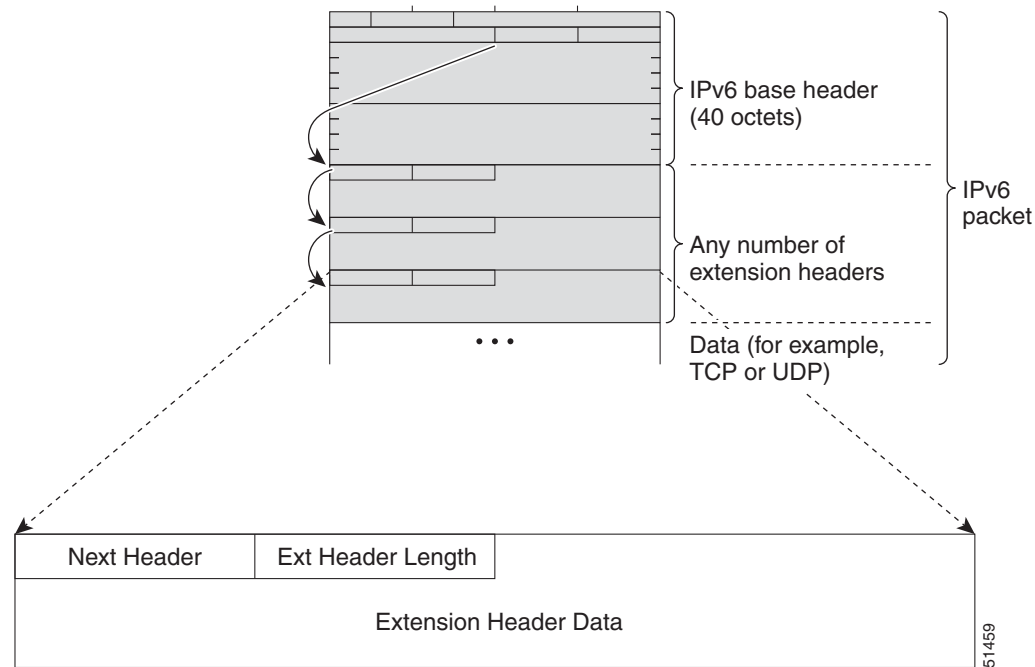
Table 3-2 Base IPv6 Packet Header Fields (continued)

Field	Description
Payload Length	Similar to the Total Length field in the IPv4 packet header. The Payload Length field indicates the total length of the data portion of the packet.
Next Header	Similar to the Protocol field in the IPv4 packet header. The value of the Next Header field determines the type of information that follows the base IPv6 header. The type of information that follows the base IPv6 header can be a transport-layer packet, for example, a TCP or UDP packet, or an Extension Header, as shown in Figure 3-6 .
Hop Limit	Similar to the Time to Live field in the IPv4 packet header. The value of the Hop Limit field specifies the maximum number of routers that an IPv6 packet can pass through before the packet is considered invalid. Each router decrements the value by one. Because no checksum is in the IPv6 header, the router can decrement the value without needing to recalculate the checksum, which saves processing resources.
Source Address	Similar to the Source Address field in the IPv4 packet header, except that the field contains a 128-bit source address for IPv6 instead of a 32-bit source address for IPv4.
Destination Address	Similar to the Destination Address field in the IPv4 packet header, except that the field contains a 128-bit destination address for IPv6 instead of a 32-bit destination address for IPv4.

Figure 3-6 IPv6 Packet Header Format

Optional extension headers and the data portion of the packet are after the eight fields of the base IPv6 packet header. If present, each extension header is aligned to 64 bits. There is no fixed number of extension headers in an IPv6 packet. Each extension header is identified by the Next Header field of the previous header. Typically, the final extension header has a Next Header field of a transport-layer protocol, such as TCP or UDP. [Figure 3-7](#) shows the IPv6 extension header format.

Figure 3-7 IPv6 Extension Header Format



[Table 3-3](#) lists the extension header types and their Next Header field values.

Table 3-3 IPv6 Extension Header Types

Header Type	Next Header Value	Description
Hop-by-hop options header	0	Header that is processed by all hops in the path of a packet. When present, the hop-by-hop options header always follows immediately after the base IPv6 packet header.
Destination options header	60	Header that can follow any hop-by-hop options header. The header is processed at the final destination and at each visited address specified by a routing header. Alternatively, the destination options header can follow any Encapsulating Security Payload (ESP) header. The destination options header is processed only at the final destination.
Routing header	43	Header that is used for source routing.

Table 3-3 IPv6 Extension Header Types (continued)

Header Type	Next Header Value	Description
Fragment header	44	Header that is used when a source fragments a packet that is larger than the maximum transmission unit (MTU) for the path between itself and a destination. The Fragment header is used in each fragmented packet.
Upper-layer headers	6 (TCP) 17 (UDP)	Headers that are used inside a packet to transport the data. The two main transport protocols are TCP and UDP.

DNS for IPv6

IPv6 supports DNS record types that are supported in the DNS name-to-address and address-to-name lookup processes. The DNS record types support IPv6 addresses (see [Table 3-4](#)).


Note

IPv6 also supports the reverse mapping of IPv6 addresses to DNS names.

Table 3-4 IPv6 DNS Record Types

Record Type	Description	Format
AAAA	Maps a hostname to an IPv6 address. (Equivalent to an A record in IPv4.)	www.abc.test AAAA 3FFE:YYYY:C18:1::2
PTR	Maps an IPv6 address to a hostname. (Equivalent to a PTR record in IPv4.)	2.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.8.1.c.0.y.y.y.e.f.f.3.ip6.int PTR www.abc.test

Path MTU Discovery for IPv6

As in IPv4, you can use path MTU discovery in IPv6 to allow a host to dynamically discover and adjust to differences in the MTU size of every link along a data path. In IPv6, however, fragmentation is handled by the source of a packet when the path MTU of one link along a given data path is not large enough to accommodate the size of the packets. Having IPv6 hosts handle packet fragmentation saves IPv6 router processing resources and helps IPv6 networks run more efficiently. Once the path MTU is reduced by the arrival of an ICMP Too Big message, Cisco NX-OS retains the lower value. The connection does not increase the segment size to gauge the throughput.


Note

In IPv6, the minimum link MTU is 1280 octets. We recommend that you use an MTU value of 1500 octets for IPv6 links.

CDP IPv6 Address Support

You can use the Cisco Discovery Protocol (CDP) IPv6 address support for the neighbor information feature to transfer IPv6 addressing information between two Cisco devices. Cisco Discovery Protocol support for IPv6 addresses provides IPv6 information to network management products and troubleshooting tools.

ICMP for IPv6

You can use ICMP in IPv6 to provide information about the health of the network. ICMPv6, the version that works with IPv6, reports errors if packets cannot be processed correctly and sends informational messages about the status of the network. For example, if a router cannot forward a packet because it is too large to be sent out on another network, the router sends out an ICMPv6 message to the originating host. Additionally, ICMP packets in IPv6 are used in IPv6 neighbor discovery and path MTU discovery. The path MTU discovery process ensures that a packet is sent using the largest possible size that is supported on a specific route.

A value of 58 in the Next Header field of the base IPv6 packet header identifies an IPv6 ICMP packet. The ICMP packet follows all the extension headers and is the last piece of information in the IPv6 packet. Within the IPv6 ICMP packets, the ICMPv6 Type and ICMPv6 Code fields identify IPv6 ICMP packet specifics, such as the ICMP message type. The value in the Checksum field is computed by the sender and checked by the receiver from the fields in the IPv6 ICMP packet and the IPv6 pseudo header.

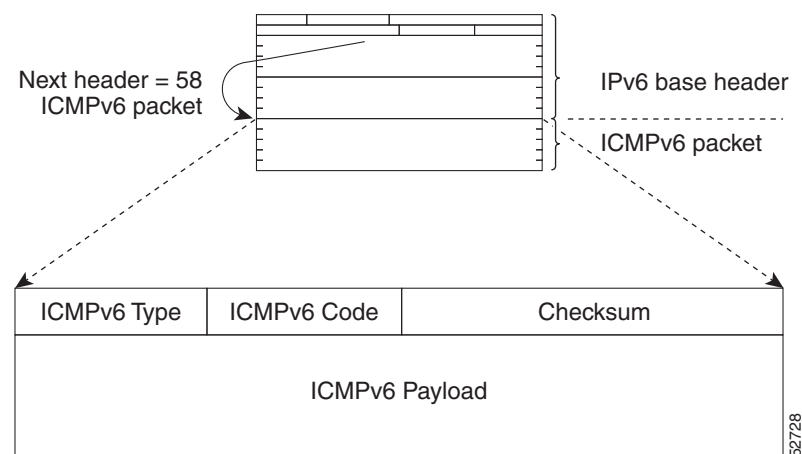


Note

The IPv6 header does not have a checksum. But a checksum on the transport layer can determine if packets have not been delivered correctly. All checksum calculations that include the IP address in the calculation must be modified for IPv6 to accommodate the new 128-bit address. A checksum is generated using a pseudo header.

The ICMPv6 Payload field contains error or diagnostic information that relates to IP packet processing. [Figure 3-8](#) shows the IPv6 ICMP packet header format.

Figure 3-8 IPv6 ICMP Packet Header Format



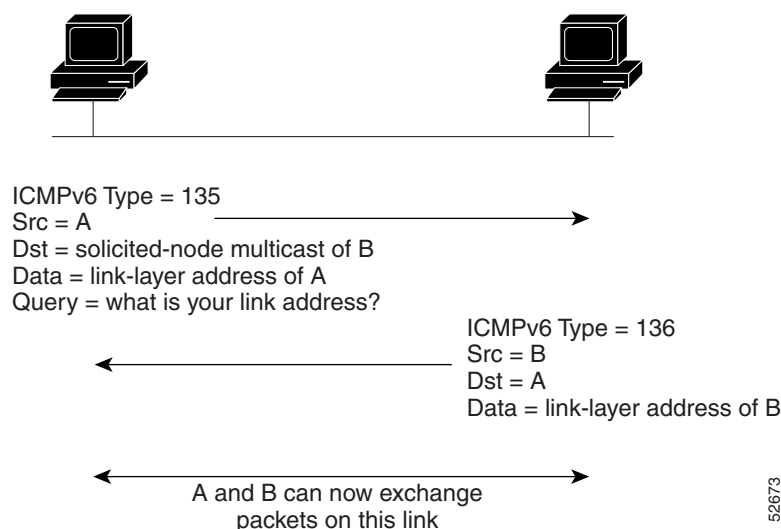
IPv6 Neighbor Discovery

You can use the IPv6 Neighbor Discovery Protocol (NDP) to determine whether a neighboring router is reachable. IPv6 nodes use neighbor discovery to determine the addresses of nodes on the same network (local link), to find neighboring routers that can forward their packets, to verify whether neighboring routers are reachable or not, and to detect changes to link-layer addresses. NDP uses ICMP messages to detect whether packets are sent to neighboring routers that are unreachable.

IPv6 Neighbor Solicitation Message

A node sends a neighbor solicitation message, which has a value of 135 in the Type field of the ICMP packet header, on the local link when it wants to determine the link-layer address of another node on the same local link (see Figure 3-9). The source address is the IPv6 address of the node that sends the neighbor solicitation message. The destination address is the solicited-node multicast address that corresponds to the IPv6 address of the destination node. The neighbor solicitation message also includes the link-layer address of the source node.

Figure 3-9 IPv6 Neighbor Discovery—Neighbor Solicitation Message



After receiving the neighbor solicitation message, the destination node replies by sending a neighbor advertisement message, which has a value of 136 in the Type field of the ICMP packet header, on the local link. The source address is the IPv6 address of the node (the IPv6 address of the node interface that sends the neighbor advertisement message). The destination address is the IPv6 address of the node that sends the neighbor solicitation message. The data portion includes the link-layer address of the node that sends the neighbor advertisement message.

After the source node receives the neighbor advertisement, the source node and destination node can communicate.

Neighbor solicitation messages can verify the reachability of a neighbor after a node identifies the link-layer address of a neighbor. When a node wants to verify the reachability of a neighbor, it uses the destination address in a neighbor solicitation message as the unicast address of the neighbor.

Neighbor advertisement messages are also sent when there is a change in the link-layer address of a node on a local link. When there is a change, the destination address for the neighbor advertisement is the all-nodes multicast address.

Neighbor unreachability detection identifies the failure of a neighbor or the failure of the forward path to the neighbor and is used for all paths between hosts and neighboring nodes (hosts or routers). Neighbor unreachability detection is performed for neighbors to which only unicast packets are being sent and is not performed for neighbors to which multicast packets are being sent.

A neighbor is considered reachable when a positive acknowledgment is returned from the neighbor (indicating that packets previously sent to the neighbor have been received and processed). A positive acknowledgment—from an upper-layer protocol (such as TCP)—indicates that a connection is making forward progress (reaching its destination). If packets are reaching the peer, they are also reaching the next-hop neighbor of the source. Forward progress is also a confirmation that the next-hop neighbor is reachable.

For destinations that are not on the local link, forward progress implies that the first-hop router is reachable. When acknowledgments from an upper-layer protocol are not available, a node probes the neighbor using unicast neighbor solicitation messages to verify that the forward path is still working. The return of a solicited neighbor advertisement message from the neighbor is a positive acknowledgment that the forward path is still working (neighbor advertisement messages that have the solicited flag set to a value of 1 are sent only in response to a neighbor solicitation message). Unsolicited messages confirm only the one-way path from the source to the destination node; solicited neighbor advertisement messages indicate that a path is working in both directions.

**Note**

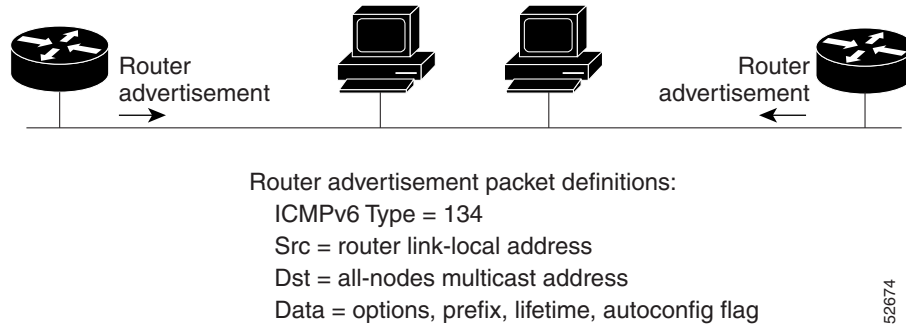
A neighbor advertisement message that has the solicited flag set to a value of 0 is not considered as a positive acknowledgment that the forward path is still working.

Neighbor solicitation messages are also used in the stateless autoconfiguration process to verify the uniqueness of unicast IPv6 addresses before the addresses are assigned to an interface. Duplicate address detection is performed first on a new, link-local IPv6 address before the address is assigned to an interface (the new address remains in a tentative state while duplicate address detection is performed). A node sends a neighbor solicitation message with an unspecified source address and a tentative link-local address in the body of the message. If another node is already using that address, the node returns a neighbor advertisement message that contains the tentative link-local address. If another node is simultaneously verifying the uniqueness of the same address, that node also returns a neighbor solicitation message. If no neighbor advertisement messages are received in response to the neighbor solicitation message and no neighbor solicitation messages are received from other nodes that are attempting to verify the same tentative address, the node that sent the original neighbor solicitation message considers the tentative link-local address to be unique and assigns the address to the interface.

IPv6 Router Advertisement Message

Router advertisement (RA) messages, which have a value of 134 in the Type field of the ICMP packet header, are periodically sent out to each configured interface of an IPv6 router. For stateless autoconfiguration to work properly, the advertised prefix length in RA messages must always be 64 bits.

The RA messages are sent to the all-nodes multicast address (see [Figure 3-10](#)).

Figure 3-10 IPv6 Neighbor Discovery—RA Message

RA messages typically include the following information:

- One or more onlink IPv6 prefixes that nodes on the local link can use to automatically configure their IPv6 addresses
- Life-time information for each prefix included in the advertisement
- Sets of flags that indicate the type of autoconfiguration (stateless or stateful) that can be completed
- Default router information (whether the router sending the advertisement should be used as a default router and, if so, the amount of time in seconds that the router should be used as a default router)
- Additional information for hosts, such as the hop limit and MTU that a host should use in packets that it originates

RAs are also sent in response to router solicitation messages. Router solicitation messages, which have a value of 133 in the Type field of the ICMP packet header, are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled RA message. The source address is usually the unspecified IPv6 address (0:0:0:0:0:0:0:0). If the host has a configured unicast address, the unicast address of the interface that sends the router solicitation message is used as the source address in the message. The destination address is the all-routers multicast address with a scope of the link. When an RA is sent in response to a router solicitation, the destination address in the RA message is the unicast address of the source of the router solicitation message.

You can configure the following RA message parameters:

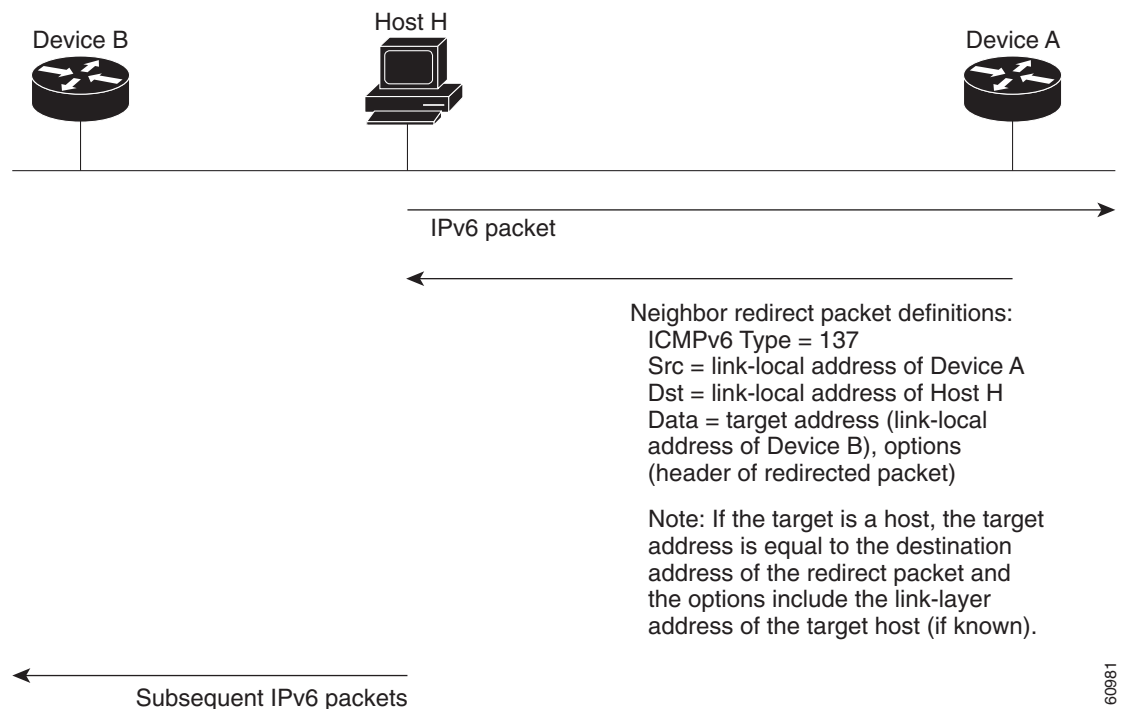
- The time interval between periodic RA messages
- The router life-time value, which indicates the usefulness of a router as the default router (for use by all nodes on a given link)
- The network prefixes in use on a given link
- The time interval between neighbor solicitation message retransmissions (on a given link)
- The amount of time that a node considers a neighbor reachable (for use by all nodes on a given link)

The configured parameters are specific to an interface. The sending of RA messages (with default values) is automatically enabled on Ethernet interfaces. For other interface types, you must enter the **no ipv6 nd suppress-ra** command to send RA messages. You can disable the RA message feature on individual interfaces by entering the **ipv6 nd suppress-ra** command.

IPv6 Neighbor Redirect Message

Routers send neighbor redirect messages to inform hosts of better first-hop nodes on the path to a destination (see [Figure 3-11](#)). A value of 137 in the Type field of the ICMP packet header identifies an IPv6 neighbor redirect message.

Figure 3-11 IPv6 Neighbor Discovery—Neighbor Redirect Message



Note

A router must be able to determine the link-local address for each of its neighboring routers in order to ensure that the target address (the final destination) in a redirect message identifies the neighbor router by its link-local address. For static routing, you should specify the address of the next-hop router using the link-local address of the router. For dynamic routing, you must configure all IPv6 routing protocols to exchange the link-local addresses of neighboring routers.

After forwarding a packet, a router sends a redirect message to the source of the packet under the following circumstances:

- The destination address of the packet is not a multicast address.
- The packet was not addressed to the router.
- The packet is about to be sent out the interface on which it was received.
- The router determines that a better first-hop node for the packet resides on the same link as the source of the packet.
- The source address of the packet is a global IPv6 address of a neighbor on the same link or a link-local address.

Virtualization Support

IPv6 supports virtual routing and forwarding (VRF) instances.

IPv6 Routes with ECMP

If all next-hops for a route are glean, drop, or punt, all next-hops are programmed as-is in the Multipath hardware table.

If some next-hops for a route are glean, drop, or punt, and the remaining next-hops are not, then only non glean, drop, or punt next-hops are programmed in the Multipath hardware table.

When a specific next-hop for ECMP route is resolved (ARP/IPV6 ND resolved), then the Multipath hardware table is updated accordingly.

Licensing Requirements for IPv6

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	IPv6 requires 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. For a complete explanation of the Cisco NX-OS licensing scheme, see the <i>Cisco NX-OS Licensing Guide</i> .

Prerequisites for IPv6

IPv6 has the following prerequisites:

- You must be familiar with IPv6 basics such as IPv6 addressing, IPv6 header information, ICMPv6, and the IPv6 Neighbor Discovery (ND) Protocol.
- Ensure that you follow the memory/processing guidelines when you make a device a dual-stack device (IPv4/IPv6).

Guidelines and Limitations for IPv6

IPv6 has the following configuration guidelines and limitations:

- IPv6 packets are transparent to Layer 2 LAN switches because the switches do not examine Layer 3 packet information before forwarding IPv6 frames. IPv6 hosts can be directly attached to Layer 2 LAN switches.
- You can configure multiple IPv6 global addresses within the same prefix on an interface. However, multiple IPv6 link-local addresses on an interface are not supported.
- Because RFC 3879 deprecates the use of site-local addresses, you should configure private IPv6 addresses according to the recommendations of unique local addressing (ULA) in RFC 4193.

Default Settings

Table 3-5 lists the default settings for IPv6 parameters.

Table 3-5 **Default IPv6 Parameters**

Parameters	Default
ND reachable time	0 milliseconds
neighbor solicitation retransmit interval	1000 milliseconds

Configuring IPv6

This section includes the following topics:

- [Configuring IPv6 Addressing, page 3-17](#)
- [Configuring the System Routing Mode, page 3-19](#)
- [Configuring ALPM, page 3-21](#)
- [Configuring IPv6 Neighbor Discovery, page 3-23](#)
- [Optional IPv6 Neighbor Discovery, page 3-25](#)
- [Configuring IPv6 Packet Verification, page 3-26](#)



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.

Configuring IPv6 Addressing

You must configure an IPv6 address on an interface so that the interface can forward IPv6 traffic. When you configure a global IPv6 address on an interface, it automatically configures a link-local address and activates IPv6 for that interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet *number***
3. **ipv6 address {*addr* [*eui64*] [*route-preference preference*] [*secondary*] tag *tag-id*}]**
or
ipv6 address *ipv6-address* use-link-local-only
4. (Optional) **show ipv6 interface**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	ipv6 address { <i>addr</i> [eui64] [route-preference <i>preference</i>] [secondary] tag <i>tag-id</i>] or ipv6 address <i>ipv6-address</i> use-link-local-only Example: switch(config-if)# ipv6 address 2001:0DB8::1/10 or switch(config-if)# ipv6 address use-link-local-only	Specifies an IPv6 address assigned to the interface and enables IPv6 processing on the interface. Entering the ipv6 address command configures global IPv6 addresses with an interface identifier (ID) in the low-order 64 bits of the IPv6 address. Only the 64-bit network prefix for the address needs to be specified; the last 64 bits are automatically computed from the interface ID. Entering the ipv6 address use-link-local-only command configures a link-local address on the interface that is used instead of the link-local address that is automatically configured when IPv6 is enabled on the interface. This command enables IPv6 processing on an interface without configuring an IPv6 address.
Step 4	show ipv6 interface Example: switch(config-if)# show ipv6 interface	(Optional) Displays interfaces configured for IPv6.
Step 5	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure an IPv6 address:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# ipv6 address ?
A:B::C:D/LEN IPv6 prefix format: xxxx:xxxx/ml, xxxx:xxxx::/ml,
xxxx::xx/128
use-link-local-only Enable IPv6 on interface using only a single link-local
address
switch(config-if)# ipv6 address 2001:db8::/64 eui64
```

This example shows how to display an IPv6 interface:

```
switch(config-if)# show ipv6 interface ethernet 3/1
Ethernet3/1, Interface status: protocol-down/link-down/admin-down, iod: 36
  IPv6 address: 0dc3:0dc3:0000:0000:0218:baff:fed8:239d
  IPv6 subnet: 0dc3:0dc3:0000:0000:0000:0000:0000:0000/64
```

```

IPv6 link-local address: fe80::0218:baff:fed8:239d (default)
IPv6 multicast routing: disabled
IPv6 multicast groups locally joined:
    ff02::0001:ffd8:239d ff02::0002 ff02::0001 ff02::0001:ffd8:239d
IPv6 multicast (S,G) entries joined: none
IPv6 MTU: 1500 (using link MTU)
IPv6 RP inbound packet-filtering policy: none
IPv6 RP outbound packet-filtering policy: none
IPv6 inbound packet-filtering policy: none
IPv6 outbound packet-filtering policy: none
IPv6 interface statistics last reset: never
IPv6 interface RP-traffic statistics: (forwarded/originated/consumed)
    Unicast packets: 0/0/0
    Unicast bytes: 0/0/0
    Multicast packets: 0/0/0
    Multicast bytes: 0/0/0

```

Configuring the System Routing Mode

By default, the switch programs routes in a manner that allows for longest prefix match (LPM) scale on the device.

You can modify the default LPM scale, according to the requirements when the node is positioned as a Layer-2 to Layer-3 boundary node.



Note You must reload the switch by using the **reload** command to apply the configured system routing mode. You cannot use the **fast-reload** command to apply the configured system routing mode.

SUMMARY STEPS

1. **configure terminal**
2. **[no] system routing max-mode l3**
3. **copy running-config startup-config**
4. **reload**
5. (Optional) **show hardware profile status**
6. (Optional) **show running-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[no] system routing max-mode l3 Example: switch(config)# system routing max-mode l3	Enables algorithmic longest prefix match (ALPM) mode. When ALPM mode is enabled, you can carve the LPM table for IPv6 routes. The no form of this command disables ALPM.

	Command	Purpose
Step 3	copy running-config startup-config Example: switch(config)# copy running-config startup-config	Saves this configuration change.
Step 4	reload Example: switch(config)# reload	Reboots the entire device.
Step 5	show hardware profile status Example: switch(config)# show hardware profile status	(Optional) Displays the updated number of LPM routes and the LPM mode.
Step 6	show running-config Example: switch(config)# show running-config	(Optional) Displays the applied configuration.

This example shows how to configure the system routing mode and enable ALPM:

```
switch# configure terminal
switch(config)# system routing max-mode 13
switch(config)# copy running-config startup-config
[#####] 100%
Copy complete, now saving to disk (please wait)...
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
switch(config)# show hardware profile status
Total LPM Entries = 65023.
Total Host Entries = 16384.
Reserved LPM Entries = 1024.
Max Host4/Host6 Limit Entries (shared)= 8192/4096*
Max Mcast Limit Entries = 4096.
Max Ucast IPv6 LPM Limit Entries = 20479.
Max Ucast IPv6 LPM_65_to_127 Limit Entries = 0.
Used LPM Entries (Total) = 55.
Used IPv4 LPM Entries = 27.
Used IPv6 LPM Entries = 28.
Used IPv6 LPM_65_to_127 Entries = 0.
Used Host Entries in LPM (Total) = 2.
Used Host4 Entries in LPM = 2.
Used Host6 Entries in LPM = 0.
Used Mcast Entries = 0.
Used Mcast OIFL Entries = 2.
Used Host Entries in Host (Total) = 152.
Used Host4 Entries in Host = 102.
Used Host6 Entries in Host = 50.
Max ECMP Table Entries = 256.
Used ECMP Table Entries = 0.
MFIB prefer-source-tree = Disabled/0/0.

*Unicast LPM Table is in ALPM mode shared b/n v4 & v6...
*Unicast Host Table is in shared mode b/n v4 & v6...
```

Configuring ALPM

You can use the 64-bit algorithmic longest prefix match (ALPM) feature to manage IPv4 and IPv6 route table entries. In ALPM mode, the device can store significantly more route entries. Without IPv6 ALPM carving, you can program the route as follows:

URPF Status	IPv4 + IPv6-64 (/0-64)
Disabled	Upto 131071 IPv4 or upto 81919IPv6 /64
Enabled	Upto 65535 IPv4 or upto 20479 IPv6 /64



Note

There is no support for IPv6-128(/65-127) routes in this mode.

With IPv6 ALPM carving, you can program the route as follows:

ipv6_alpm_carve_value	IPv4 (non-URPF)/(URPF)	IPv6-64(/0-64) + IPv6-128(/65-127)
1024	98303/unsupported	5119/unsupported
2048	65535/32767	10239/2047
3072	32767/unsupported	15359/unsupported

All IPv6 routes including /65-127b routes share the same table. The IPv4 routes have dedicated memory.



Note

In ALPM mode, only upto 40K IPv4 + 40K IPv6 with 16 way ECMP is supported with ECMP routes.

After you carve the LPM table, the carved space is used for all IPv6 routes with masks between /0 and /127. The remaining space is used only for IPv4 routes.

Without explicit IPv6 carving, the system supports IPv6 routes with masks between /0 and /64.



Note

You must reload the switch by using the **reload** command to apply the configured carving values. You cannot use the **fast-reload** command to apply the configured carving values.

SUMMARY STEPS

1. **configure terminal**
2. **[no] hardware profile ipv6 alpm carve-value** *ipv6_alpm_carve_value*
3. **copy running-config startup-config**
4. **reload**
5. (Optional) **show hardware profile status**
6. (Optional) **show running-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[no] hardware profile ipv6 alpm carve-value <i>ipv6_alpm_carve_value</i> Example: switch(config)# hardware profile ipv6 alpm 2048	Enables /128 IPv6 routes and carves the LPM table for all IPv6 entries.
Step 3	copy running-config startup-config Example: switch(config)# copy running-config startup-config	Saves this configuration change.
Step 4	reload Example: switch(config)# reload	Reboots the entire device.
Step 5	show hardware profile status Example: switch(config)# show hardware profile status	(Optional) Displays the updated number of LPM routes and the LPM mode.
Step 6	show running-config Example: switch(config)# show running-config	(Optional) Displays the applied configuration.

**Note**

The carve-value 1024 and 3072 are not supported when urpf is enabled.

This example shows how to configure LPM table carving:

```
switch# configure terminal
switch(config)# hardware profile ipv6 alpm carve-value 2048
Warning: Please copy running-config to startup-config and reload the switch to apply
changes
switch(config)# copy running-config startup-config
[#####] 100%
Copy complete, now saving to disk (please wait)...
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
switch(config)# show hardware profile status
Max Ucast IPv4 LPM Limit Entries = 65535.
Total Host Entries = 16384.
Reserved LPM Entries = 1024.
Max Host4/Host6 Limit Entries (shared)= 8192/4096*
Max Mcast Limit Entries = 4096.
Max Ucast IPv6 LPM Limit Entries = 10239.
Used IPv4 LPM Entries = 65533.
Used IPv6 LPM Entries = 10239.
```



```

Used Host Entries in LPM (Total) = 2.
Used Host4 Entries in LPM = 2.
Used Host6 Entries in LPM = 0.
Used Mcast Entries = 0.
Used Mcast OIFL Entries = 2.
Used Host Entries in Host (Total) = 21.
Used Host4 Entries in Host = 19.
Used Host6 Entries in Host = 2.
Max ECMP Table Entries = 256.
Used ECMP Table Entries = 0.
MFIB prefer-source-tree = Disabled/0/0.

*Unicast LPM Table is in ALPM mode with ipv6 carving...
*IPv6 LPM and LPM_65_TO_127 tables are in shared mode...
*Unicast Host Table is in shared mode b/n v4 & v6...

```

Configuring IPv6 Neighbor Discovery

You can configure IPv6 neighbor discovery on the router. NDP enables IPv6 nodes and routers to determine the link-layer address of a neighbor on the same link, find neighboring routers, and keep track of neighbors.

BEFORE YOU BEGIN

You must first enable IPv6 on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet *number***
3. **ipv6 nd [hop-limit *hop-limit* | managed-config-flag | mtu *mtu* | ns-interval *interval* | other-config-flag | prefix | ra-interval *interval* | ra-lifetime *lifetime* | reachable-time *time* | redirects | retrans-timer *time* | suppress-ra]**
4. (Optional) **show ipv6 nd interface**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/31 switch(config-if)#	Enters interface configuration mode.

	Command	Purpose
Step 3	<pre> ipv6 nd [hop-limit <i>hop-limit</i> managed-config-flag mtu <i>mtu</i> ns-interval <i>interval</i> other-config-flag prefix ra-interval <i>interval</i> ra-lifetime <i>lifetime</i> reachable-time <i>time</i> redirects retrans-timer <i>time</i> suppress-ra] Example: switch(config-if)# ipv6 nd prefix </pre>	<p>Neighbor discovery is enabled automatically when you configure an IPv6 address. This command enables the following additional IPv6 neighbor discovery options on the interface:</p> <ul style="list-style-type: none"> • hop-limit <i>hop-limit</i>—Advertises the hop limit in IPv6 neighbor discovery packets. The range is from 0 to 255. • managed-config-flag—Advertises in ICMPv6 router-advertisement messages to use stateful address auto configuration to obtain address information. • mtu <i>mtu</i>—Advertises the maximum transmission unit (MTU) in ICMPv6 router-advertisement messages on this link. The range is from 1280 to 65535 bytes. • ns-interval <i>interval</i>—Configures the retransmission interval between IPv6 neighbor solicitation messages. The range is from 1000 to 3600000 milliseconds. • other-config-flag—Indicates in ICMPv6 router-advertisement messages that hosts use stateful auto configuration to obtain nonaddress related information. • prefix—Advertises the IPv6 prefix in the router-advertisement messages. • ra-interval <i>interval</i>—Configures the interval between sending ICMPv6 router-advertisement messages. The range is from 4 to 1800 seconds. • ra-lifetime <i>lifetime</i>—Advertises the lifetime of a default router in ICMPv6 router-advertisement messages. The range is from 0 to 9000 seconds. • reachable-time <i>time</i>—Advertises the time when a node considers a neighbor up after receiving a reachability confirmation in ICMPv6 router-advertisement messages. The range is from 0 to 9000 seconds. • redirects—Enables sending ICMPv6 redirect messages. • retrans-timer <i>time</i>—Advertises the time between neighbor-solicitation messages in ICMPv6 router-advertisement messages. The range is from 0 to 9000 seconds. • suppress-ra—Disables sending ICMPv6 router-advertisement messages.

	Command	Purpose
Step 4	show ipv6 nd interface Example: switch(config-if)# show ipv6 nd interface	(Optional) Displays interfaces configured for IPv6 neighbor discovery.
Step 5	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure IPv6 neighbor discovery reachable time:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# ipv6 nd reachable-time 10
```

This example shows how to display an IPv6 neighbor discovery interface:

```
switch(config-if)# show ipv6 nd interface ethernet 3/1
ICMPv6 ND Interfaces for VRF "default"
Ethernet3/1, Interface status: protocol-down/link-down/admin-down
IPv6 address: 0dc3:0dc3:0000:0000:0218:baff:fed8:239d
  ICMPv6 active timers:
    Last Neighbor-Solicitation sent: never
    Last Neighbor-Advertisement sent: never
    Last Router-Advertisement sent: never
    Next Router-Advertisement sent in: 0.000000
  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: 10 ms
    Send "Retrans Timer" field: 0 ms
  Neighbor-Solicitation parameters:
    NS retransmit interval: 1000 ms
  ICMPv6 error message parameters:
    Send redirects: false
    Send unreachable: false
```

Optional IPv6 Neighbor Discovery

You can use the following optional IPv6 Neighbor Discovery commands:

Command	Purpose
ipv6 nd hop-limit	Configures the maximum number of hops used in router advertisements and all IPv6 packets that are originated by the router.
ipv6 nd managed-config-flag	Sets the managed address configuration flag in IPv6 router advertisements.

Command	Purpose
ipv6 nd mtu	Sets the maximum transmission unit (MTU) size of IPv6 packets sent on an interface.
ipv6 nd ns-interval	Configures the interval between IPv6 neighbor solicitation retransmissions on an interface.
ipv6 nd other-config-flag	Configures the other stateful configuration flag in IPv6 router advertisements.
ipv6 nd ra-interval	Configures the interval between IPv6 router advertisement (RA) transmissions on an interface.
ipv6 nd ra-lifetime	Configures the router lifetime value in IPv6 router advertisements on an interface.
ipv6 nd reachable-time	Configures the amount of time that a remote IPv6 node is considered reachable after some reachability confirmation event has occurred.
ipv6 nd redirects	Enables ICMPv6 redirect messages to be sent.
ipv6 nd retrans-timer	Configures the advertised time between neighbor solicitation messages in router advertisements.
ipv6 nd suppress-ra	Suppresses IPv6 router advertisement transmissions on a LAN interface.

Configuring IPv6 Packet Verification

Cisco NX-OS supports an Intrusion Detection System (IDS) that checks for IPv6 packet verification. You can enable or disable these IDS checks.

To enable IDS checks, use the following commands in global configuration mode:

Command	Purpose
hardware ip verify address {destination zero identical reserved source multicast}	<p>Performs the following IDS checks on the IPv6 address:</p> <ul style="list-style-type: none"> • destination zero—Drops IPv6 packets if the destination IP address is ::. • identical—Drops IPv6 packets if the source IPv6 address is identical to the destination IPv6 address. • reserved—Drops IPv6 packets if the IPv6 address is ::1. • source multicast—Drops IPv6 packets if the IPv6 source address is in the FF00::/8 range (multicast).

Command	Purpose
hardware ipv6 verify length {consistent maximum {max-frag max-tcp udp}}	Performs the following IDS checks on the IPv6 address: <ul style="list-style-type: none"> • consistent—Drops IPv6 packets where the Ethernet frame size is greater than or equal to the IPv6 packet length plus the Ethernet header. • maximum max-frag—Drops IPv6 packets if the formula (IPv6 Payload Length – IPv6 Extension Header Bytes) + (Fragment Offset * 8) is greater than 65536. • maximum max-tcp—Drops IPv6 packets if the TCP length is greater than the IP payload length. • maximum udp—Drops IPv6 packets if the IPv6 payload length is less than the UDP packet length.
hardware ipv6 verify tcp tiny-frag	Drops TCP packets if the IPv6 fragment offset is 1, or if the IPv6 fragment offset is 0 and the IP payload length is less than 16.
hardware ipv6 verify version	Drops IPv6 packets if the EtherType is not set to 6 (IPv6).

Use the **show hardware forwarding ip verify** command to display the IPv6 packet verification configuration.

Verifying the IPv6 Configuration

To display the IPv6 configuration, perform one of the following tasks:

Command	Purpose
show hardware forwarding ip verify	Displays the IPv4 and IPv6 packet verification configuration.
show ipv6 interface	Displays IPv6-related interface information.
show ipv6 adjacency	Displays the adjacency table.
show ipv6 icmp	Displays ICMPv6 information.
show ipv6 nd	Displays IPv6 neighbor discovery interface information.
show ipv6 neighbor	Displays IPv6 neighbor entry.

Configuration Examples for IPv6

This example shows how to configure IPv6:

```
configure terminal
interface ethernet 3/1
  ipv6 address 2001:db8::/64 eui64
  ipv6 nd reachable-time 10
```

Additional References

For additional information related to implementing IPv6, see the following sections:

- [Related Documents, page 3-28](#)
- [Standards, page 3-28](#)

Related Documents

Related Topic	Document Title
IPv6 CLI commands	<i>Cisco Nexus 3000 Series NX-OS Unicast Routing Command Reference, Release 6.x</i>

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—



PART 2

Routing



Configuring OSPFv2

This chapter describes how to configure Open Shortest Path First version 2 (OSPFv2) for IPv4 networks.

This chapter includes the following sections:

- [Information About OSPFv2, page 4-1](#)
- [Licensing Requirements for OSPFv2, page 4-12](#)
- [Prerequisites for OSPFv2, page 4-12](#)
- [Default Settings, page 4-12](#)
- [Guidelines and Limitations, page 4-12](#)
- [Configuring Basic OSPFv2, page 4-13](#)
- [Configuring Advanced OSPFv2, page 4-22](#)
- [Verifying the OSPFv2 Configuration, page 4-41](#)
- [Displaying OSPFv2 Statistics, page 4-42](#)
- [Configuration Examples for OSPFv2, page 4-42](#)
- [Additional References, page 4-43](#)

Information About OSPFv2

OSPFv2 is an IETF link-state protocol (see the [“Link-State Protocols” section on page 1-9](#)) for IPv4 networks. An OSPFv2 router sends a special message, called a *hello packet*, out each OSPF-enabled interface to discover other OSPFv2 neighbor routers. Once a neighbor is discovered, the two routers compare information in the hello packet to determine if the routers have compatible configurations. The neighbor routers attempt to establish *adjacency*, which means that the routers synchronize their link-state databases to ensure that they have identical OSPFv2 routing information. Adjacent routers share *link-state advertisements* (LSAs) that include information about the operational state of each link, the cost of the link, and any other neighbor information. The routers then flood these received LSAs out every OSPF-enabled interface so that all OSPFv2 routers eventually have identical link-state databases. When all OSPFv2 routers have identical link-state databases, the network is *converged* (see the [“Convergence” section on page 1-6](#)). Each router then uses Dijkstra’s Shortest Path First (SPF) algorithm to build its route table.

You can divide OSPFv2 networks into areas. Routers send most LSAs only within one area, which reduces the CPU and memory requirements for an OSPF-enabled router.

OSPFv2 supports IPv4.

This section includes the following topics:

- [Hello Packet, page 4-2](#)
- [Neighbors, page 4-2](#)
- [Adjacency, page 4-3](#)
- [Designated Routers, page 4-3](#)
- [Areas, page 4-4](#)
- [Link-State Advertisements, page 4-5](#)
- [OSPFv2 and the Unicast RIB, page 4-7](#)
- [Authentication, page 4-7](#)
- [Advanced Features, page 4-8](#)

Hello Packet

OSPFv2 routers periodically send hello packets on every OSPF-enabled interface. The *hello interval* determines how frequently the router sends these hello packets and is configured per interface. OSPFv2 uses hello packets for the following tasks:

- Neighbor discovery
- Keepalives
- Designated router election (see the [“Designated Routers” section on page 4-3](#))

The hello packet contains information about the originating OSPFv2 interface and router, including the assigned OSPFv2 cost of the link, the hello interval, and optional capabilities of the originating router. An OSPFv2 interface that receives these hello packets determines if the settings are compatible with the receiving interface settings. Compatible interfaces are considered neighbors and are added to the neighbor table (see the [“Neighbors” section on page 4-2](#)).

Hello packets also include a list of router IDs for the routers that the originating interface has communicated with. If the receiving interface sees its own router ID in this list, then bidirectional communication has been established between the two interfaces.

OSPFv2 uses hello packets as a keepalive message to determine if a neighbor is still communicating. If a router does not receive a hello packet by the configured *dead interval* (usually a multiple of the hello interval), then the neighbor is removed from the local neighbor table.

Neighbors

An OSPFv2 interface must have a compatible configuration with a remote interface before the two can be considered neighbors. The two OSPFv2 interfaces must match the following criteria:

- Hello interval
- Dead interval
- Area ID (see the [“Areas” section on page 4-4](#))
- Authentication
- Optional capabilities

If there is a match, the following information is entered into the neighbor table:

- Neighbor ID—The router ID of the neighbor.
- Priority—Priority of the neighbor. The priority is used for designated router election (see the [“Designated Routers” section on page 4-3](#)).
- State—Indication of whether the neighbor has just been heard from, is in the process of setting up bidirectional communications, is sharing the link-state information, or has achieved full adjacency.
- Dead time—Indication of the time since the last Hello packet was received from this neighbor.
- IP Address—The IP address of the neighbor.
- Designated Router—Indication of whether the neighbor has been declared as the designated router or as the backup designated router (see the [“Designated Routers” section on page 4-3](#)).
- Local interface—The local interface that received the hello packet for this neighbor.

Adjacency

Not all neighbors establish adjacency. Depending on the network type and designated router establishment, some neighbors become fully adjacent and share LSAs with all their neighbors, while other neighbors do not. For more information, see the [“Designated Routers” section on page 4-3](#).

Adjacency is established using Database Description packets, Link State Request packets, and Link State Update packets in OSPF. The Database Description packet includes only the LSA headers from the link-state database of the neighbor (see the [“Link-State Database” section on page 4-7](#)). The local router compares these headers with its own link-state database and determines which LSAs are new or updated. The local router sends a Link State Request packet for each LSA that it needs new or updated information on. The neighbor responds with a Link State Update packet. This exchange continues until both routers have the same link-state information.

Designated Routers

Networks with multiple routers present a unique situation for OSPF. If every router floods the network with LSAs, the same link-state information will be sent from multiple sources. Depending on the type of network, OSPFv2 might use a single router, the *designated router* (DR), to control the LSA floods and represent the network to the rest of the OSPFv2 area (see the [“Areas” section on page 4-4](#)). If the DR fails, OSPFv2 selects a *backup designated router* (BDR). If the DR fails, OSPFv2 uses the BDR.

Network types are as follows:

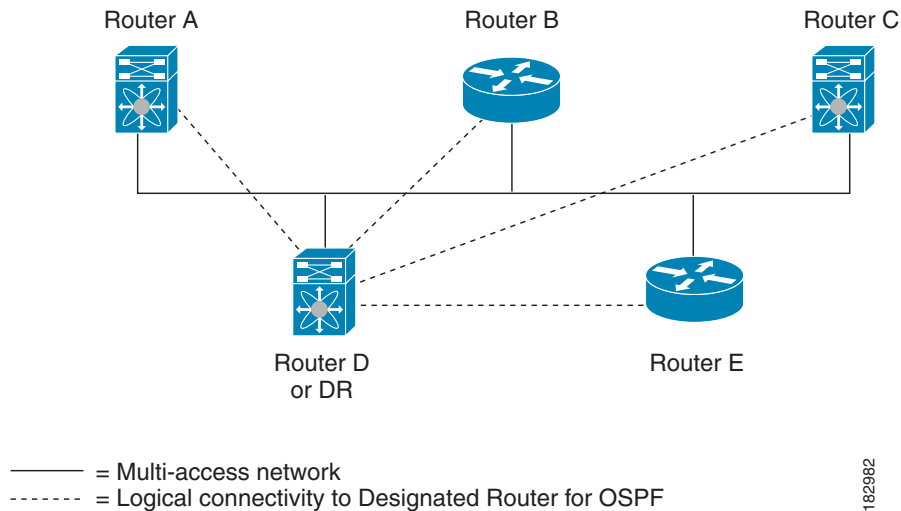
- Point-to-point—A network that exists only between two routers. All neighbors on a point-to-point network establish adjacency and there is no DR.
- Broadcast—A network with multiple routers that can communicate over a shared medium that allows broadcast traffic, such as Ethernet. OSPFv2 routers establish a DR and BDR that controls LSA flooding on the network. OSPFv2 uses a MAC address of 0100.5300.0005 to communicate with neighbors.

The DR and BDR are selected based on the information in the Hello packet. When an interface sends a Hello packet, it sets the priority field and the DR and BDR field if it knows who the DR and BDR are. The routers follow an election procedure based on which routers declare themselves in the DR and BDR fields and the priority field in the Hello packet. As a final tie breaker, OSPFv2 chooses the highest router IDs as the DR and BDR.

All other routers establish adjacency with the DR and the BDR. [Figure 4-1](#) shows this adjacency relationship between all routers and the DR.

DRs are based on a router interface. A router might be the DR for one network and not for another network on a different interface.

Figure 4-1 DR in Multi-Access Network

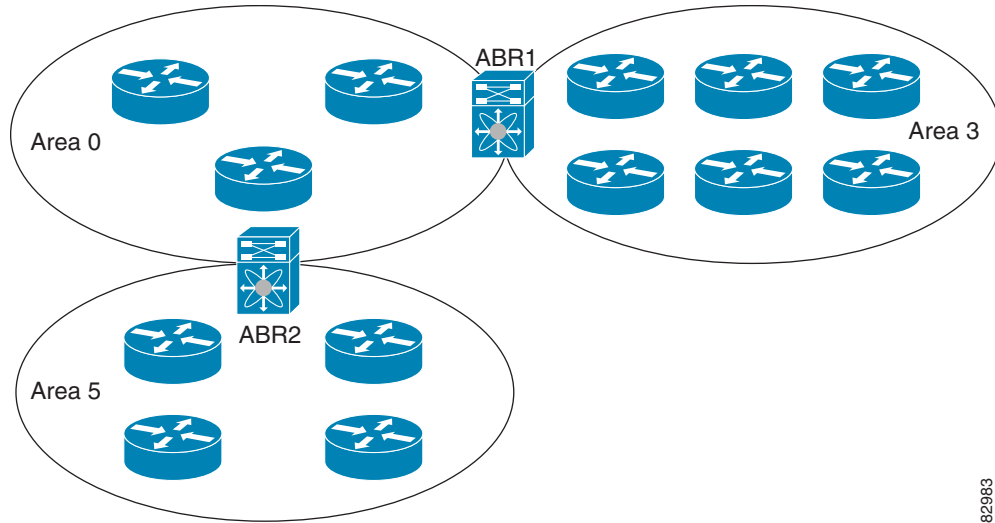


Areas

You can limit the CPU and memory requirements that OSPFv2 puts on the routers by dividing an OSPFv2 network into *areas*. An area is a logical division of routers and links within an OSPFv2 domain that creates separate subdomains. LSA flooding is contained within an area, and the link-state database is limited to links within the area. You can assign an area ID to the interfaces within the defined area. The Area ID is a 32-bit value that you can enter as a number or in dotted decimal notation, such as 10.2.3.1.

Cisco NX-OS always displays the area in dotted decimal notation.

If you define more than one area in an OSPFv2 network, you must also define the backbone area, which has the reserved area ID of 0. If you have more than one area, then one or more routers become *area border routers* (ABRs). An ABR connects to both the backbone area and at least one other defined area (see Figure 4-2).

Figure 4-2 OSPFv2 Areas

182983

The ABR has a separate link-state database for each area to which it connects. The ABR sends Network Summary (type 3) LSAs (see the [“Route Summarization”](#) section on page 4-10) from one connected area to the backbone area. The backbone area sends summarized information about one area to another area. In [Figure 4-2](#), Area 0 sends summarized information about Area 5 to Area 3.

OSPFv2 defines one other router type: the autonomous system boundary router (ASBR). This router connects an OSPFv2 area to another autonomous system. An autonomous system is a network controlled by a single technical administration entity. OSPFv2 can redistribute its routing information into another autonomous system or receive redistributed routes from another autonomous system. For more information, see [“Advanced Features”](#) section on page 4-8.)

Link-State Advertisements

OSPFv2 uses link-state advertisements (LSAs) to build its routing table.

This section includes the following topics:

- [LSA Types, page 4-5](#)
- [Link Cost, page 4-6](#)
- [Flooding and LSA Group Pacing, page 4-6](#)
- [Link-State Database, page 4-7](#)
- [Opaque LSAs, page 4-7](#)

LSA Types

[Table 4-1](#) shows the LSA types supported by Cisco NX-OS.

Table 4-1 LSA Types

Type	Name	Description
1	Router LSA	LSA sent by every router. This LSA includes the state and the cost of all links and a list of all OSPFv2 neighbors on the link. Router LSAs trigger an SPF recalculation. Router LSAs are flooded to local OSPFv2 area.
2	Network LSA	LSA sent by the DR. This LSA lists all routers in the multi-access network. Network LSAs trigger an SPF recalculation. See the “Designated Routers” section on page 4-3 .
3	Network Summary LSA	LSA sent by the area border router to an external area for each destination in the local area. This LSA includes the link cost from the area border router to the local destination. See the “Areas” section on page 4-4 .
4	ASBR Summary LSA	LSA sent by the area border router to an external area. This LSA advertises the link cost to the ASBR only. See the “Areas” section on page 4-4 .
5	AS External LSA	LSA generated by the ASBR. This LSA includes the link cost to an external autonomous system destination. AS External LSAs are flooded throughout the autonomous system. See the “Areas” section on page 4-4 .
7	NSSA External LSA	LSA generated by the ASBR within a not-so-stubby area (NSSA). This LSA includes the link cost to an external autonomous system destination. NSSA External LSAs are flooded only within the local NSSA. See the “Areas” section on page 4-4 .
9–11	Opaque LSAs	LSA used to extend OSPF. See the “Opaque LSAs” section on page 4-7 .

Link Cost

Each OSPFv2 interface is assigned a *link cost*. The cost is an arbitrary number. By default, Cisco NX-OS assigns a cost that is the configured reference bandwidth divided by the interface bandwidth. By default, the reference bandwidth is 40 Gb/s. The link cost is carried in the LSA updates for each link.

Flooding and LSA Group Pacing

When an OSPFv2 router receives an LSA, it forwards that LSA out every OSPF-enabled interface, flooding the OSPFv2 area with this information. This LSA flooding guarantees that all routers in the network have identical routing information. LSA flooding depends on the OSPFv2 area configuration (see the [“Areas” section on page 4-4](#)). The LSAs are flooded based on the *link-state refresh* time (every 30 minutes by default). Each LSA has its own link-state refresh time.

You can control the flooding rate of LSA updates in your network by using the LSA group pacing feature. LSA group pacing can reduce high CPU or buffer utilization. This feature groups LSAs with similar link-state refresh times to allow OSPFv2 to pack multiple LSAs into an OSPFv2 Update message.

By default, LSAs with link-state refresh times within four minutes of each other are grouped together. You should lower this value for large link-state databases or raise it for smaller databases to optimize the OSPFv2 load on your network.

Link-State Database

Each router maintains a link-state database for the OSPFv2 network. This database contains all the collected LSAs, and includes information on all the routes through the network. OSPFv2 uses this information to calculate the best path to each destination and populates the routing table with these best paths.

LSAs are removed from the link-state database if no LSA update has been received within a set interval, called the MaxAge. Routers flood a repeat of the LSA every 30 minutes to prevent accurate link-state information from being aged out. Cisco NX-OS supports the LSA grouping feature to prevent all LSAs from refreshing at the same time. For more information, see the [“Flooding and LSA Group Pacing” section on page 4-6](#).

Opaque LSAs

Opaque LSAs allow you to extend OSPF functionality. Opaque LSAs consist of a standard LSA header followed by application-specific information. This information might be used by OSPFv2 or by other applications. Three Opaque LSA types are defined as follows:

- LSA type 9—Flooded to the local network.
- LSA type 10—Flooded to the local area.
- LSA type 11—Flooded to the local autonomous system.

OSPFv2 and the Unicast RIB

OSPFv2 runs the Dijkstra shortest path first algorithm on the link-state database. This algorithm selects the best path to each destination based on the sum of all the link costs for each link in the path. The resultant shortest path for each destination is then put in the OSPFv2 route table. When the OSPFv2 network is converged, this route table feeds into the unicast RIB. OSPFv2 communicates with the unicast RIB to do the following:

- Add or remove routes
- Handle route redistribution from other protocols
- Provide convergence updates to remove stale OSPFv2 routes and for stub router advertisements (see the [“OSPFv2 Stub Router Advertisements” section on page 4-11](#))

OSPFv2 also runs a modified Dijkstra algorithm for fast recalculation for summary and external (type 3, 4, 5, and 7) LSA changes.

Authentication

You can configure authentication on OSPFv2 messages to prevent unauthorized or invalid routing updates in your network. Cisco NX-OS supports two authentication methods:

- Simple password authentication
- MD5 authentication digest

You can configure the OSPFv2 authentication for an OSPFv2 area or per interface.

Simple Password Authentication

Simple password authentication uses a simple cleartext password that is sent as part of the OSPFv2 message. The receiving OSPFv2 router must be configured with the same cleartext password to accept the OSPFv2 message as a valid route update. Because the password is in cleartext, anyone who can watch traffic on the network can learn the password.

MD5 Authentication

You should use MD5 authentication to authenticate OSPFv2 messages. You configure a password that is shared at the local router and all remote OSPFv2 neighbors. For each OSPFv2 message, Cisco NX-OS creates an MD5 one-way message digest based on the message itself and the encrypted password. The interface sends this digest with the OSPFv2 message. The receiving OSPFv2 neighbor validates the digest using the same encrypted password. If the message has not changed, the digest calculation is identical and the OSPFv2 message is considered valid.

MD5 authentication includes a sequence number with each OSPFv2 message to ensure that no message is replayed in the network.

Advanced Features

Cisco NX-OS supports a number of advanced OSPFv2 features that enhance the usability and scalability of OSPFv2 in the network. This section includes the following topics:

- [Stub Area, page 4-8](#)
- [Not-So-Stubby Area, page 4-9](#)
- [Virtual Links, page 4-9](#)
- [Route Redistribution, page 4-10](#)
- [Route Summarization, page 4-10](#)
- [OSPFv2 Stub Router Advertisements, page 4-11](#)
- [Multiple OSPFv2 Instances, page 4-11](#)
- [SPF Optimization, page 4-11](#)
- [BFD, page 4-11](#)
- [Virtualization Support, page 4-12](#)

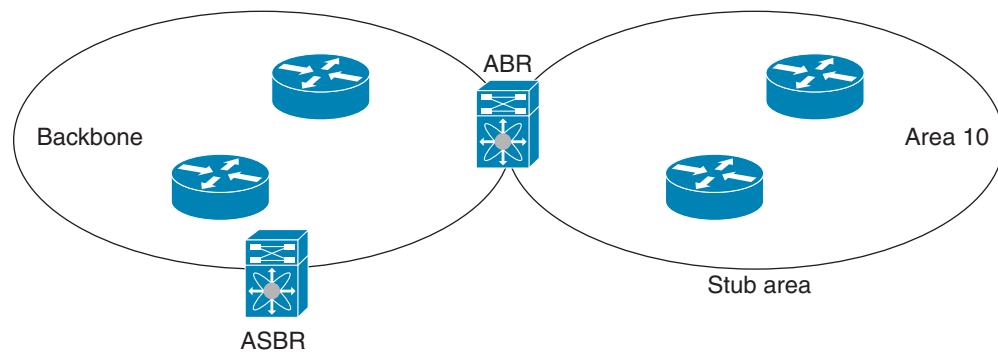
Stub Area

You can limit the amount of external routing information that floods an area by making it a *stub area*. A stub area is an area that does not allow AS External (type 5) LSAs (see the [“Link-State Advertisements” section on page 4-5](#)). These LSAs are usually flooded throughout the local autonomous system to propagate external route information. Stub areas have the following requirements:

- All routers in the stub area are stub routers. See the [“Stub Routing” section on page 1-7](#).
- No ASBR routers exist in the stub area.
- You cannot configure virtual links in the stub area.

Figure 4-3 shows an example of an OSPFv2 autonomous system where all routers in area 0.0.0.10 have to go through the ABR to reach external autonomous systems. area 0.0.0.10 can be configured as a stub area.

Figure 4-3 Stub Area



Stub areas use a default route for all traffic that needs to go through the backbone area to the external autonomous system. The default route is 0.0.0.0 for IPv4.

Not-So-Stubby Area

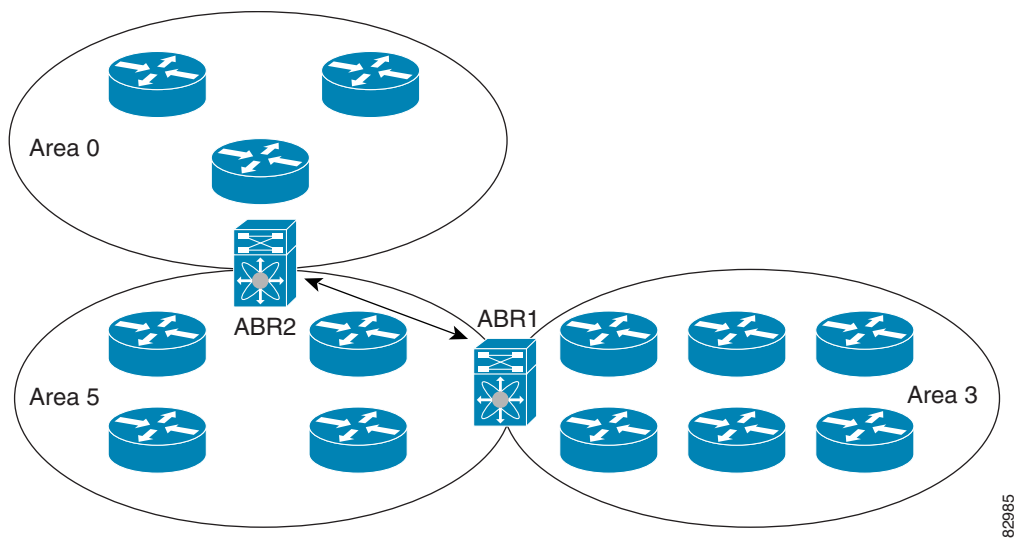
A Not-so-Stubby Area ([NSSA](#)) is similar to a stub area, except that an NSSA allows you to import autonomous system external routes within an NSSA using redistribution. The NSSA ASBR redistributes these routes and generates NSSA External (type 7) LSAs that it floods throughout the NSSA. You can optionally configure the ABR that connects the NSSA to other areas to translate this NSSA External LSA to AS External (type 5) LSAs. The area border router (ABR) then floods these AS External LSAs throughout the OSPFv2 autonomous system. Summarization and filtering are supported during the translation. See the [“Link-State Advertisements”](#) section on page 4-5 for details on NSSA External LSAs.

You can, for example, use NSSA to simplify administration if you are connecting a central site using OSPFv2 to a remote site that is using a different routing protocol. Before NSSA, the connection between the corporate site border router and a remote router could not be run as an OSPFv2 stub area because routes for the remote site could not be redistributed into a stub area. With NSSA, you can extend OSPFv2 to cover the remote connection by defining the area between the corporate router and remote router as an NSSA (see the [“Configuring NSSA”](#) section on page 4-26).

The backbone Area 0 cannot be an NSSA.

Virtual Links

Virtual links allow you to connect an OSPFv2 area ABR to a backbone area ABR when a direct physical connection is not available. Figure 4-4 shows a virtual link that connects Area 3 to the backbone area through Area 5.

Figure 4-4 Virtual Links

You can also use virtual links to temporarily recover from a partitioned area, which occurs when a link within the area fails, isolating part of the area from reaching the designated ABR to the backbone area.

Route Redistribution

OSPFv2 can learn routes from other routing protocols by using route redistribution. See the [“Route Redistribution” section on page 1-6](#). You configure OSPFv2 to assign a link cost for these redistributed routes or a default link cost for all redistributed routes.

Route redistribution uses route maps to control which external routes are redistributed. See [Chapter 16, “Configuring Route Policy Manager,”](#) for details on configuring route maps. You can use route maps to modify parameters in the AS External (type 5) and NSSA External (type 7) LSAs before these external routes are advertised in the local OSPFv2 autonomous system.

Route Summarization

Because OSPFv2 shares all learned routes with every OSPF-enabled router, you might want to use route summarization to reduce the number of unique routes that are flooded to every OSPF-enabled router. Route summarization simplifies route tables by replacing more-specific addresses with an address that represents all the specific addresses. For example, you can replace 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24 with one summary address, 10.1.0.0/16.

Typically, you would summarize at the boundaries of area border routers (ABRs). Although you could configure summarization between any two areas, it is better to summarize in the direction of the backbone so that the backbone receives all the aggregate addresses and injects them, already summarized, into other areas. The two types of summarization are as follows:

- Inter-area route summarization
- External route summarization

You configure inter-area route summarization on ABRs, summarizing routes between areas in the autonomous system. To take advantage of summarization, you should assign network numbers in areas in a contiguous way to be able to lump these addresses into one range.

External route summarization is specific to external routes that are injected into OSPFv2 using route redistribution. You should make sure that external ranges that are being summarized are contiguous. Summarizing overlapping ranges from two different routers could cause packets to be sent to the wrong destination. Configure external route summarization on ASBRs that are redistributing routes into OSPF.

When you configure a summary address, Cisco NX-OS automatically configures a discard route for the summary address to prevent routing black holes and route loops.

OSPFv2 Stub Router Advertisements

You can configure an OSPFv2 interface to act as a stub router using the OSPFv2 stub router advertisements feature. Use this feature when you want to limit the OSPFv2 traffic through this router, such as when you want to introduce a new router to the network in a controlled manner or limit the load on a router that is already overloaded. You might also want to use this feature for various administrative or traffic engineering reasons.

OSPFv2 stub router advertisements do not remove the OSPFv2 router from the network topology, but they do prevent other OSPFv2 routers from using this router to route traffic to other parts of the network. Only the traffic that is destined for this router or directly connected to this router is sent.

OSPFv2 stub router advertisements mark all stub links (directly connected to the local router) to the cost of the local OSPFv2 interface. All remote links are marked with the maximum cost (0xFFFF).

Multiple OSPFv2 Instances

Cisco NX-OS supports multiple instances of the OSPFv2 protocol that run on the same node. You cannot configure multiple instances over the same interface. By default, every instance uses the same system router ID. You must manually configure the router ID for each instance if the instances are in the same OSPFv2 autonomous system.

SPF Optimization

Cisco NX-OS optimizes the SPF algorithm in the following ways:

- Partial SPF for Network (type 2) LSAs, Network Summary (type 3) LSAs, and AS External (type 5) LSAs—When there is a change on any of these LSAs, Cisco NX-OS performs a faster partial calculation rather than running the whole SPF calculation.
- SPF timers—You can configure different timers for controlling SPF calculations. These timers include exponential backoff for subsequent SPF calculations. The exponential backoff limits the CPU load of multiple SPF calculations.

BFD

This feature supports bidirectional forwarding detection (BFD). BFD is a detection protocol that provides fast forwarding-path failure detection times. 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.

Virtualization Support

OSPFv2 supports Virtual Routing and Forwarding (VRF) instances. By default, Cisco NX-OS places you in the default VRF unless you specifically configure another VRF. Each OSPFv2 instance can support multiple VRFs, up to the system limit. For more information, see [Chapter 14, “Configuring Layer 3 Virtualization.”](#)

Licensing Requirements for OSPFv2

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	OSPFv2 requires a LAN Base Services license. For over 256 routes, OSPFv2 requires an Enterprise services license. For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the <i>Cisco NX-OS Licensing Guide</i> .

Prerequisites for OSPFv2

OSPFv2 has the following prerequisites:

- You must be familiar with routing fundamentals to configure OSPF.
- You are logged on to the switch.
- You have configured at least one interface for IPv4 that is capable of communicating with a remote OSPFv2 neighbor.
- You have installed the LAN Base Services license.
- You have completed the OSPFv2 network strategy and planning for your network. For example, you must decide whether multiple areas are required.
- You have enabled the OSPF feature (see the [“Enabling the OSPFv2 Feature”](#) section on page 4-13).

Guidelines and Limitations

OSPFv2 has the following configuration guidelines and limitations:

- Cisco NX-OS displays areas in dotted decimal notation regardless of whether you enter the area in decimal or dotted decimal notation.

**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.

Default Settings

[Table 4-2](#) lists the default settings for OSPFv2 parameters.

Table 4-2 **Default OSPFv2 Parameters**

Parameters	Default
Hello interval	10 seconds
Dead interval	40 seconds
OSPFv2 feature	Disabled
Stub router advertisement announce time	600 seconds
Reference bandwidth for link cost calculation	40 Gb/s
LSA minimal arrival time	1000 milliseconds
LSA group pacing	240 seconds
SPF calculation initial delay time	200 milliseconds
SPF calculation minimum hold time	1000 milliseconds
SPF calculation maximum wait time	5000 milliseconds

Configuring Basic OSPFv2

Configure OSPFv2 after you have designed your OSPFv2 network.

This section includes the following topics:

- [Enabling the OSPFv2 Feature, page 4-13](#)
- [Creating an OSPFv2 Instance, page 4-14](#)
- [Configuring Optional Parameters on an OSPFv2 Instance, page 4-15](#)
- [Configuring Networks in OSPFv2, page 4-16](#)
- [Configuring Authentication for an Area, page 4-18](#)
- [Configuring Authentication for an Interface, page 4-20](#)

Enabling the OSPFv2 Feature

You must enable the OSPFv2 feature before you can configure OSPFv2.

SUMMARY STEPS

1. **configure terminal**
2. **feature ospf**
3. (Optional) **show feature**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	feature ospf Example: switch(config)# feature ospf	Enables the OSPFv2 feature.
Step 3	show feature Example: switch(config)# show feature	(Optional) Displays enabled and disabled features.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

Use the **no feature ospf** command to disable the OSPFv2 feature and remove all associated configurations.

Command	Purpose
no feature ospf Example: switch(config)# no feature ospf	Disables the OSPFv2 feature and removes all associated configurations.

Creating an OSPFv2 Instance

The first step in configuring OSPFv2 is to create an OSPFv2 instance. You assign a unique instance tag for this OSPFv2 instance. The instance tag can be any string.

For more information about OSPFv2 instance parameters, see the [“Configuring Advanced OSPFv2” section on page 4-22](#).

BEFORE YOU BEGIN

Ensure that you have enabled the OSPF feature (see the [“Enabling the OSPFv2 Feature” section on page 4-13](#)).

Use the **show ip ospf instance-tag** command to verify that the instance tag is not in use.

OSPFv2 must be able to obtain a router identifier (for example, a configured loopback address) or you must configure the router ID option.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf instance-tag**

3. (Optional) **router-id** *ip-address*
4. (Optional) **show ip ospf** *instance-tag*
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf <i>instance-tag</i> Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	router-id <i>ip-address</i> Example: switch(config-router)# router-id 192.0.2.1	(Optional) Configures the OSPFv2 router ID. This IP address identifies this OSPFv2 instance and must exist on a configured interface in the system.
Step 4	show ip ospf <i>instance-tag</i> Example: switch(config-router)# show ip ospf 201	(Optional) Displays OSPF information.
Step 5	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

Use the **no router ospf** command to remove the OSPFv2 instance and all associated configurations.

Command	Purpose
no router ospf <i>instance-tag</i> Example: switch(config)# no router ospf 201	Deletes the OSPF instance and the associated configurations.



Note

This command does not remove OSPF configuration in interface mode. You must manually remove any OSPFv2 commands configured in interface mode.

Configuring Optional Parameters on an OSPFv2 Instance

You can configure optional parameters for OSPF.

For more information about OSPFv2 instance parameters, see the [“Configuring Advanced OSPFv2” section on page 4-22](#).

BEFORE YOU BEGIN

Ensure that you have enabled the OSPF feature (see the [“Enabling the OSPFv2 Feature”](#) section on page 4-13).

OSPFv2 must be able to obtain a router identifier (for example, a configured loopback address) or you must configure the router ID option.

DETAILED STEPS

Command	Purpose
distance <i>number</i> Example: switch(config-router)# distance 25	Configures the administrative distance for this OSPFv2 instance. The range is from 1 to 255. The default is 110.
log-adjacency-changes [detail] Example: switch(config-router)# log-adjacency-changes	Generates a system message whenever a neighbor changes state.
maximum-paths <i>path-number</i> Example: switch(config-router)# maximum-paths 4	Configures the maximum number of equal OSPFv2 paths to a destination in the route table. This command is used for load balancing. The range is from 1 to 16. The default is 8.

This example shows how to create an OSPFv2 instance:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# copy running-config startup-config
```

Configuring Networks in OSPFv2

You can configure a network to OSPFv2 by associating it through the interface that the router uses to connect to that network (see the [“Neighbors”](#) section on page 4-2). You can add all networks to the default backbone area (Area 0), or you can create new areas using any decimal number or an IP address.



Note

All areas must connect to the backbone area either directly or through a virtual link.



Note

OSPF is not enabled on an interface until you configure a valid IP address for that interface.

BEFORE YOU BEGIN

Ensure that you have enabled the OSPF feature (see the [“Enabling the OSPFv2 Feature”](#) section on page 4-13).

SUMMARY STEPS

1. **configure terminal**

2. **interface** *interface-type slot/port*
3. **no switchport**
4. **ip address** *ip-prefix/length*
5. **ip router ospf instance-tag area area-id** [secondaries none]
6. (Optional) **show ip ospf instance-tag interface interface-type slot/port**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	ip address <i>ip-prefix/length</i> Example: switch(config-if)# ip address 192.0.2.1/16	Assigns an IP address and subnet mask to this interface.
Step 5	ip router ospf instance-tag area area-id [secondaries none] Example: switch(config-if)# ip router ospf 201 area 0.0.0.15	Adds the interface to the OSPFv2 instance and area.
Step 6	show ip ospf instance-tag interface <i>interface-type slot/port</i> Example: switch(config-if)# show ip ospf 201 interface ethernet 1/2	(Optional) Displays OSPF information.
Step 7	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

You can configure the following optional parameters for OSPFv2 in interface configuration mode:

Command	Purpose
ip ospf cost <i>number</i> Example: switch(config-if)# ip ospf cost 25	Configures the OSPFv2 cost metric for this interface. The default is to calculate cost metric, based on reference bandwidth and interface bandwidth. The range is from 1 to 65535.
ip ospf dead-interval <i>seconds</i> Example: switch(config-if)# ip ospf dead-interval 50	Configures the OSPFv2 dead interval, in seconds. The range is from 1 to 65535. The default is four times the hello interval, in seconds.
ip ospf hello-interval <i>seconds</i> Example: switch(config-if)# ip ospf hello-interval 25	Configures the OSPFv2 hello interval, in seconds. The range is from 1 to 65535. The default is 10 seconds.
ip ospf mtu-ignore Example: switch(config-if)# ip ospf mtu-ignore	Configures OSPFv2 to ignore any IP MTU mismatch with a neighbor. The default is to not establish adjacency if the neighbor MTU does not match the local interface MTU.
ip ospf passive-interface Example: switch(config-if)# ip ospf passive-interface	Suppresses routing updates on the interface.
ip ospf priority <i>number</i> Example: switch(config-if)# ip ospf priority 25	Configures the OSPFv2 priority, used to determine the DR for an area. The range is from 0 to 255. The default is 1. See the “Designated Routers” section on page 4-3.
ip ospf shutdown Example: switch(config-if)# ip ospf shutdown	Shuts down the OSPFv2 instance on this interface.

This example shows how to add a network area 0.0.0.10 in OSPFv2 instance 201:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# ip address 192.0.2.1/16
switch(config-if)# ip router ospf 201 area 0.0.0.10
switch(config-if)# copy running-config startup-config
```

Use the **show ip ospf interface** command to verify the interface configuration. Use the **show ip ospf neighbor** command to see the neighbors for this interface.

Configuring Authentication for an Area

You can configure authentication for all networks in an area or for individual interfaces in the area. Interface authentication configuration overrides area authentication.

BEFORE YOU BEGIN

Ensure that you have enabled the OSPF feature (see the [“Enabling the OSPFv2 Feature”](#) section on page 4-13).

Ensure that all neighbors on an interface share the same authentication configuration, including the shared authentication key.

Create the keychain for this authentication configuration.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **area** *area-id* **authentication** [**message-digest**]
4. **interface** *interface-type slot/port*
5. **no switchport**
6. (Optional) **ip ospf authentication-key** [**0** | **3**] *key*
or
(Optional) **ip ospf message-digest-key** *key-id* **md5** [**0** | **3**] *key*
7. (Optional) **show ip ospf** *instance-tag* **interface** *interface-type slot/port*
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf <i>instance-tag</i> Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	area <i>area-id</i> authentication [message-digest] Example: switch(config-router)# area 0.0.0.10 authentication	Configures the authentication mode for an area.
Step 4	interface <i>interface-type slot/port</i> Example: switch(config-router)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 5	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.

	Command	Purpose
Step 6	ip ospf authentication-key [0 3] <i>key</i> Example: switch(config-if)# ip ospf authentication-key 0 mypass	(Optional) Configures simple password authentication for this interface. Use this command if the authentication is not set to keychain or message-digest. 0 configures the password in cleartext. 3 configures the password as 3DES encrypted.
	ip ospf message-digest-key <i>key-id</i> md5 [0 3] <i>key</i> Example: switch(config-if)# ip ospf message-digest-key 21 md5 0 mypass	(Optional) Configures message digest authentication for this interface. Use this command if the authentication is set to message-digest. The key-id range is from 1 to 255. The MD5 option 0 configures the password in cleartext and 3 configures the pass key as 3DES encrypted.
Step 7	show ip ospf instance-tag interface <i>interface-type slot/port</i> Example: switch(config-if)# show ip ospf 201 interface ethernet 1/2	(Optional) Displays OSPF information.
Step 8	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

Configuring Authentication for an Interface

You can configure authentication for individual interfaces in the area. Interface authentication configuration overrides area authentication.

BEFORE YOU BEGIN

Ensure that you have enabled the OSPF feature (see the [“Enabling the OSPFv2 Feature”](#) section on page 4-13).

Ensure that all neighbors on an interface share the same authentication configuration, including the shared authentication key.

Create the keychain for this authentication configuration.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **no switchport**
4. **ip ospf authentication** [message-digest]
5. (Optional) **ip ospf authentication keychain** *key-name*
6. (Optional) **ip ospf authentication-key** [0 | 3 | 7] *key*
7. (Optional) **ip ospf message-digest-key** *key-id* md5 [0 | 3 | 7] *key*
8. (Optional) **show ip ospf instance-tag interface** *interface-type slot/port*
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface interface-type slot/port Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	ip ospf authentication [message-digest] Example: switch(config-if)# ip ospf authentication	Enables interface authentication mode for OSPFv2 for either cleartext or message-digest type. Overrides area-based authentication for this interface. All neighbors must share this authentication type.
Step 5	ip ospf authentication keychain key-name Example: switch(config-if)# ip ospf authentication keychain Test1	(Optional) Configures interface authentication to use keychains for OSPFv2.
Step 6	ip ospf authentication-key [0 3 7] key Example: switch(config-if)# ip ospf authentication-key 0 mypass	(Optional) Configures simple password authentication for this interface. Use this command if the authentication is not set to keychain or message-digest. The options are as follows: <ul style="list-style-type: none"> • 0—Configures the password in cleartext. • 3—Configures the pass key as 3DES encrypted. • 7—Configures the key as Cisco type 7 encrypted.
Step 7	ip ospf message-digest-key key-id md5 [0 3 7] key Example: switch(config-if)# ip ospf message-digest-key 21 md5 0 mypass	(Optional) Configures message digest authentication for this interface. Use this command if the authentication is set to message-digest. The key-id range is from 1 to 255. The MD5 options are as follows: <ul style="list-style-type: none"> • 0—Configures the password in cleartext. • 3—Configures the pass key as 3DES encrypted. • 7—Configures the key as Cisco type 7 encrypted.

	Command	Purpose
Step 8	show ip ospf instance-tag interface <i>interface-type slot/port</i> Example: switch(config-if)# show ip ospf 201 interface ethernet 1/2	(Optional) Displays OSPF information.
Step 9	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to set an interface for simple, unencrypted passwords and set the password for Ethernet interface 1/2:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# ip router ospf 201 area 0.0.0.10
switch(config-if)# ip ospf authentication
switch(config-if)# ip ospf authentication-key 0 mypass
switch(config-if)# copy running-config startup-config
```

Configuring Advanced OSPFv2

Configure OSPFv2 after you have designed your OSPFv2 network.

This section includes the following topics:

- [Configuring Filter Lists for Border Routers, page 4-23](#)
- [Configuring Stub Areas, page 4-24](#)
- [Configuring a Totally Stubby Area, page 4-25](#)
- [Configuring NSSA, page 4-26](#)
- [Configuring Virtual Links, page 4-28](#)
- [Configuring Redistribution, page 4-30](#)
- [Limiting the Number of Redistributed Routes, page 4-32](#)
- [Configuring Route Summarization, page 4-34](#)
- [Configuring Stub Route Advertisements, page 4-35](#)
- [Modifying the Default Timers, page 4-36](#)
- [Restarting an OSPFv2 Instance, page 4-39](#)
- [Configuring OSPFv2 with Virtualization, page 4-39](#)

Configuring Filter Lists for Border Routers

You can separate your OSPFv2 domain into a series of areas that contain related networks. All areas must connect to the backbone area through an area border router (ABR). OSPFv2 domains can connect to external domains through an *autonomous system border router* (ASBR). See the “Areas” section on page 4-4.

ABRs have the following optional configuration parameters:

- Area range—Configures route summarization between areas.
- Filter list—Filters the Network Summary (type 3) LSAs on an ABR that are allowed in from an external area.

ASBRs also support filter lists.

BEFORE YOU BEGIN

Ensure that you have enabled the OSPF feature (see the “Enabling the OSPFv2 Feature” section on page 4-13).

Create the route map that the filter list uses to filter IP prefixes in incoming or outgoing Network Summary (type 3) LSAs. See Chapter 16, “Configuring Route Policy Manager.”

SUMMARY STEPS

1. **configure terminal**
2. **router ospf instance-tag**
3. **area area-id filter-list route-map map-name {in | out}**
4. (Optional) **show ip ospf policy statistics area id filter-list {in | out}**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf instance-tag Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	area area-id filter-list route-map map-name {in out} Example: switch(config-router)# area 0.0.0.10 filter-list route-map FilterLSAs in	Filters incoming or outgoing Network Summary (type 3) LSAs on an ABR.

	Command	Purpose
Step 4	show ip ospf policy statistics area <i>id</i> filter-list {in out} Example: switch(config-if)# show ip ospf policy statistics area 0.0.0.10 filter-list in	(Optional) Displays OSPF policy information.
Step 5	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure a filter list in area 0.0.0.10:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 filter-list route-map FilterLSAs in
switch(config-router)# copy running-config startup-config
```

Configuring Stub Areas

You can configure a stub area for part of an OSPFv2 domain where external traffic is not necessary. Stub areas block AS External (type 5) LSAs, limiting unnecessary routing to and from selected networks. See the “[Stub Area](#)” section on page 4-8. You can optionally block all summary routes from going into the stub area.

BEFORE YOU BEGIN

Ensure that you have enabled the OSPF feature (see the “[Enabling the OSPFv2 Feature](#)” section on page 4-13).

Ensure that there are no virtual links or ASBRs in the proposed stub area.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf *instance-tag***
3. **area *area-id* stub**
4. (Optional) **area *area-id* default-cost *cost***
5. (Optional) **show ip ospf *instance-tag***
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf instance-tag Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	area area-id stub Example: switch(config-router)# area 0.0.0.10 stub	Creates this area as a stub area.
Step 4	area area-id default-cost cost Example: switch(config-router)# area 0.0.0.10 default-cost 25	(Optional) Sets the cost metric for the default summary route sent into this stub area. The range is from 0 to 16777215. The default is 1.
Step 5	show ip ospf instance-tag Example: switch(config-if)# show ip ospf 201	(Optional) Displays OSPF information.
Step 6	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to create a stub area:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 stub
switch(config-router)# copy running-config startup-config
```

Configuring a Totally Stubby Area

You can create a totally stubby area and prevent all summary route updates from going into the stub area.

To create a totally stubby area, use the following command in router configuration mode:

Command	Purpose
area <i>area-id</i> stub no-summary	Creates this area as a totally stubby area.
Example: switch(config-router)# area 20 stub no-summary	

Configuring NSSA

You can configure an NSSA for part of an OSPFv2 domain where limited external traffic is required. See the [“Not-So-Stubby Area” section on page 4-9](#). You can optionally translate this external traffic to an AS External (type 5) LSA and flood the OSPFv2 domain with this routing information. An NSSA can be configured with the following optional parameters:

- **No redistribution**—Redistributed routes bypass the NSSA and are redistributed to other areas in the OSPFv2 autonomous system. Use this option when the NSSA ASBR is also an ABR.
- **Default information originate**—Generates an NSSA External (type 7) LSA for a default route to the external autonomous system. Use this option on an NSSA ASBR if the ASBR contains the default route in the routing table. This option can be used on an NSSA ABR whether or not the ABR contains the default route in the routing table.
- **Route map**—Filters the external routes so that only those routes that you want are flooded throughout the NSSA and other areas.
- **Translate**—Translates NSSA External LSAs to AS External LSAs for areas outside the NSSA. Use this command on an NSSA ABR to flood the redistributed routes throughout the OSPFv2 autonomous system. You can optionally suppress the forwarding address in these AS External LSAs. If you choose this option, the forwarding address is set to 0.0.0.0.
- **No summary**—Blocks all summary routes from flooding the NSSA. Use this option on the NSSA ABR.

BEFORE YOU BEGIN

Ensure that you have enabled the OSPF feature (see the [“Enabling the OSPFv2 Feature” section on page 4-13](#)).

Ensure that there are no virtual links in the proposed NSSA and that it is not the backbone area.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **area** *area-id* **nssa** [**no-redistribution**] [**default-information-originate** [**route-map** *map-name*]] [**no-summary**] [**translate type7** {**always** | **never**}] [**suppress-fa**]
4. (Optional) **area** *area-id* **default-cost** *cost*
5. (Optional) **show ip ospf** *instance-tag*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf instance-tag Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	area area-id nssa [no-redistribution] [default-information-originate] [route-map map-name]] [no-summary] [translate type7 {always never}] [suppress-fa]] Example: switch(config-router)# area 0.0.0.10 nssa	Creates this area as an NSSA.
Step 4	area area-id default-cost cost Example: switch(config-router)# area 0.0.0.10 default-cost 25	(Optional) Sets the cost metric for the default summary route sent into this NSSA.
Step 5	show ip ospf instance-tag Example: switch(config-if)# show ip ospf 201	(Optional) Displays OSPF information.
Step 6	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to create an NSSA that blocks all summary route updates:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 nssa no-summary
switch(config-router)# copy running-config startup-config
```

This example shows how to create an NSSA that generates a default route:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 nssa default-info-originate
switch(config-router)# copy running-config startup-config
```

This example shows how to create an NSSA that filters external routes and blocks all summary route updates:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 nssa route-map ExternalFilter no-summary
switch(config-router)# copy running-config startup-config
```

This example shows how to create an NSSA that always translates NSSA External (type 5) LSAs to AS External (type 7) LSAs:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 nssa translate type 7 always
switch(config-router)# copy running-config startup-config
```

Configuring Virtual Links

A virtual link connects an isolated area to the backbone area through an intermediate area. See the [“Virtual Links” section on page 4-9](#). You can configure the following optional parameters for a virtual link:

- Authentication—Sets a simple password or MD5 message digest authentication and associated keys.
- Dead interval—Sets the time that a neighbor waits for a Hello packet before declaring the local router as dead and tearing down adjacencies.
- Hello interval—Sets the time between successive Hello packets.
- Retransmit interval—Sets the estimated time between successive LSAs.
- Transmit delay—Sets the estimated time to transmit an LSA to a neighbor.



Note

You must configure the virtual link on both routers involved before the link becomes active.

You cannot add a virtual link to a stub area.

BEFORE YOU BEGIN

Ensure that you have enabled the OSPF feature (see the [“Enabling the OSPFv2 Feature” section on page 4-13](#)).

SUMMARY STEPS

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **area** *area-id* **virtual-link** *router-id*
4. (Optional) **show ip ospf virtual-link** [**brief**]
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf instance-tag Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	area area-id virtual-link router-id Example: switch(config-router)# area 0.0.0.10 virtual-link 10.1.2.3 switch(config-router-vlink)#	Creates one end of a virtual link to a remote router. You must create the virtual link on that remote router to complete the link.
Step 4	show ip ospf virtual-link [brief] Example: switch(config-router-vlink)# show ip ospf virtual-link	(Optional) Displays OSPF virtual link information.
Step 5	copy running-config startup-config Example: switch(config-router-vlink)# copy running-config startup-config	(Optional) Saves this configuration change.

You can configure the following optional commands in virtual link configuration mode:

Commands	Purpose
authentication [keychain key-id message-digest null] Example: switch(config-router-vlink)# authentication message-digest	(Optional) Overrides area-based authentication for this virtual link.
authentication-key [0 3] key Example: switch(config-router-vlink)# authentication-key 0 mypass	(Optional) Configures a simple password for this virtual link. Use this command if the authentication is not set to keychain or message-digest. 0 configures the password in cleartext. 3 configures the password as 3DES encrypted.
dead-interval seconds Example: switch(config-router-vlink)# dead-interval 50	(Optional) Configures the OSPFv2 dead interval, in seconds. The range is from 1 to 65535. The default is four times the hello interval, in seconds.
hello-interval seconds Example: switch(config-router-vlink)# hello-interval 25	(Optional) Configures the OSPFv2 hello interval, in seconds. The range is from 1 to 65535. The default is 10 seconds.

Commands	Purpose
message-digest-key <i>key-id md5 [0 3]</i> <i>key</i> Example: switch(config-router-vlink)# message-digest-key 21 md5 0 mypass	(Optional) Configures message digest authentication for this virtual link. Use this command if the authentication is set to message-digest. 0 configures the password in cleartext. 3 configures the pass key as 3DES encrypted.
retransmit-interval <i>seconds</i> Example: switch(config-router-vlink)# retransmit-interval 50	(Optional) Configures the OSPFv2 retransmit interval, in seconds. The range is from 1 to 65535. The default is 5.
transmit-delay <i>seconds</i> Example: switch(config-router-vlink)# transmit-delay 2	(Optional) Configures the OSPFv2 transmit-delay, in seconds. The range is from 1 to 450. The default is 1.

This example shows how to create a simple virtual link between two ABRs.

The configuration for ABR 1 (router ID 27.0.0.55) is as follows:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 virtual-link 10.1.2.3
switch(config-router-vlink)# copy running-config startup-config
```

The configuration for ABR 2 (Router ID 10.1.2.3) is as follows:

```
switch# configure terminal
switch(config)# router ospf 101
switch(config-router)# area 0.0.0.10 virtual-link 27.0.0.55
switch(config-router-vlink)# copy running-config startup-config
```

Configuring Redistribution

You can redistribute routes learned from other routing protocols into an OSPFv2 autonomous system through the ASBR.

You can configure the following optional parameters for route redistribution in OSPF:

- Default information originate—Generates an AS External (type 5) LSA for a default route to the external autonomous system.



Note Default information originate ignores **match** statements in the optional route map.

- Default metric—Sets all redistributed routes to the same cost metric.



Note

If you redistribute static routes, Cisco NX-OS also redistributes the default static route.

BEFORE YOU BEGIN

Ensure that you have enabled the OSPF feature (see the [“Enabling the OSPFv2 Feature”](#) section on page 4-13).

Create the necessary route maps used for redistribution.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **redistribute** {*bgp id* | *direct* | *eigrp id* | *ospf id* | *rip id* | *static*} **route-map** *map-name*
4. **default-information originate** [*always*] [*route-map map-name*]
5. **default-metric** *cost*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf <i>instance-tag</i> Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	redistribute { <i>bgp id</i> <i>direct</i> <i>eigrp id</i> <i>ospf id</i> <i>rip id</i> <i>static</i> } route-map <i>map-name</i> Example: switch(config-router)# redistribute bgp 64496 route-map FilterExternalBGP	Redistributes the selected protocol into OSPF through the configured route map. Note If you redistribute static routes, Cisco NX-OS also redistributes the default static route.
Step 4	default-information originate [<i>always</i>] [<i>route-map map-name</i>] Example: switch(config-router)# default-information-originate route-map DefaultRouteFilter	Creates a default route into this OSPF domain if the default route exists in the RIB. Use the following optional keywords: <ul style="list-style-type: none">• always —Always generates the default route of 0.0.0. even if the route does not exist in the RIB.• route-map—Generates the default route if the route map returns true. Note This command ignores match statements in the route map.
Step 5	default-metric <i>cost</i> Example: switch(config-router)# default-metric 25	Sets the cost metric for the redistributed routes. This does not apply to directly connected routes. Use a route map to set the default metric for directly connected routes.
Step 6	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to redistribute the Border Gateway Protocol (BGP) into OSPF:

```
switch# configure terminal  
switch(config)# router ospf 201  
switch(config-router)# redistribute bgp route-map FilterExternalBGP  
switch(config-router)# copy running-config startup-config
```

Limiting the Number of Redistributed Routes

Route redistribution can add many routes to the OSPFv2 route table. You can configure a maximum limit to the number of routes accepted from external protocols. OSPFv2 provides the following options to configure redistributed route limits:

- **Fixed limit**—Logs a message when OSPFv2 reaches the configured maximum. OSPFv2 does not accept any more redistributed routes. You can optionally configure a threshold percentage of the maximum where OSPFv2 will log a warning when that threshold is passed.
- **Warning only**—Logs a warning only when OSPFv2 reaches the maximum. OSPFv2 continues to accept redistributed routes.
- **Withdraw**—Starts the timeout period when OSPFv2 reaches the maximum. After the timeout period, OSPFv2 requests all redistributed routes if the current number of redistributed routes is less than the maximum limit. If the current number of redistributed routes is at the maximum limit, OSPFv2 withdraws all redistributed routes. You must clear this condition before OSPFv2 accepts more redistributed routes.

You can optionally configure the timeout period.

BEFORE YOU BEGIN

Ensure that you have enabled the OSPF feature (see the [“Enabling the OSPFv2 Feature”](#) section on page 4-13).

SUMMARY STEPS

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **redistribute** {**bgp** *id* | **direct** | **eigrp** *id* | **ospf** *id* | **rip** *id* | **static**} **route-map** *map-name*
4. **redistribute maximum-prefix** *max* [*threshold*] [**warning-only** | **withdraw** [*num-retries* *timeout*]]
5. (Optional) **show running-config ospf**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf instance-tag Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	redistribute {bgp id direct eigrp id ospf id rip id static} route-map map-name Example: switch(config-router)# redistribute bgp route-map FilterExternalBGP	Redistributes the selected protocol into OSPF through the configured route map.
Step 4	redistribute maximum-prefix max [threshold] [warning-only withdraw [num-retries timeout]] Example: switch(config-router)# redistribute maximum-prefix 1000 75 warning-only	Specifies a maximum number of prefixes that OSPFv2 will distribute. The range is from 0 to 65536. Optionally specifies the following: <ul style="list-style-type: none"> • threshold—Percent of maximum prefixes that will trigger a warning message. • warning-only—Logs an warning message when the maximum number of prefixes is exceeded. • withdraw—Withdraws all redistributed routes. Optionally tries to retrieve the redistributed routes. The <i>num-retries</i> range is from 1 to 12. The <i>timeout</i> is 60 to 600 seconds. The default is 300 seconds. Use clear ip ospf redistribution if all routes are withdrawn.
Step 5	show running-config ospf Example: switch(config-router)# show running-config ospf	(Optional) Displays the OSPFv2 configuration.
Step 6	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to limit the number of redistributed routes into OSPF:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# redistribute bgp route-map FilterExternalBGP
switch(config-router)# redistribute maximum-prefix 1000 75
```

Configuring Route Summarization

You can configure route summarization for inter-area routes by configuring an address range that is summarized. You can also configure route summarization for external, redistributed routes by configuring a summary address for those routes on an ASBR. See the [“Route Summarization” section on page 4-10](#).

BEFORE YOU BEGIN

Ensure that you have enabled the OSPF feature (see the [“Enabling the OSPFv2 Feature” section on page 4-13](#)).

SUMMARY STEPS

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **area** *area-id* **range** *ip-prefix/length* [**no-advertise**] [**cost** *cost*]
4. **summary-address** *ip-prefix/length* [**no-advertise** | **tag** *tag-id*]
5. (Optional) **show ip ospf summary-address**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf <i>instance-tag</i> Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	area <i>area-id</i> range <i>ip-prefix/length</i> [no-advertise] [cost <i>cost</i>] Example: switch(config-router)# area 0.0.0.10 range 10.3.0.0/16	Creates a summary address on an ABR for a range of addresses and optionally does not advertise this summary address in a Network Summary (type 3) LSA. The <i>cost</i> range is from 0 to 16777215.
Step 4	summary-address <i>ip-prefix/length</i> [no-advertise tag <i>tag-id</i>] Example: switch(config-router)# summary-address 10.5.0.0/16 tag 2	Creates a summary address on an ASBR for a range of addresses and optionally assigns a tag for this summary address that can be used for redistribution with route maps.

	Command	Purpose
Step 5	show ip ospf summary-address Example: switch(config-router)# show ip ospf summary-address	(Optional) Displays information about OSPF summary addresses.
Step 6	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to create summary addresses between areas on an ABR:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 range 10.3.0.0/16
switch(config-router)# copy running-config startup-config
```

This example shows how to create summary addresses on an ASBR;

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# summary-address 10.5.0.0/16
switch(config-router)# copy running-config startup-config
```

Configuring Stub Route Advertisements

Use stub route advertisements when you want to limit the OSPFv2 traffic through this router for a short time. See the [“OSPFv2 Stub Router Advertisements”](#) section on page 4-11.

Stub route advertisements can be configured with the following optional parameters:

- On startup—Sends stub route advertisements for the specified announce time.
- Wait for BGP—Sends stub router advertisements until BGP converges.

BEFORE YOU BEGIN

Ensure that you have enabled the OSPF feature (see the [“Enabling the OSPFv2 Feature”](#) section on page 4-13).

SUMMARY STEPS

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **max-metric router-lsa** [**on-startup** *announce-time*] [**wait-for bgp** *tag*]
4. (Optional) **copy running-config startup-config**



Note

You should not save the running configuration of a router when it is configured for a graceful shutdown because the router will continue to advertise a maximum metric after it is reloaded.

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf instance-tag Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	max-metric router-lsa [on-startup [announce-time] [wait-for bgp tag]] Example: switch(config-router)# max-metric router-lsa	Configures OSPFv2 stub route advertisements.
Step 4	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to enable the stub router advertisements feature on startup for the default 600 seconds:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# max-metric router-lsa on-startup
switch(config-router)# copy running-config startup-config
```

Modifying the Default Timers

OSPFv2 includes a number of timers that control the behavior of protocol messages and shortest path first (SPF) calculations. OSPFv2 includes the following optional timer parameters:

- LSA arrival time—Sets the minimum interval allowed between LSAs arriving from a neighbor. LSAs that arrive faster than this time are dropped.
- Pacing LSAs—Set the interval at which LSAs are collected into a group and refreshed, checksummed, or aged. This timer controls how frequently LSA updates occur and optimizes how many are sent in an LSA update message (see the [“Flooding and LSA Group Pacing”](#) section on page 4-6).
- Throttle LSAs—Set rate limits for generating LSAs. This timer controls how frequently an LSA is generated if no topology change occurs.
- Throttle SPF calculation—Controls how frequently the SPF calculation is run.

At the interface level, you can also control the following timers:

- Retransmit interval—Sets the estimated time between successive LSAs.
- Transmit delay—Sets the estimated time to transmit an LSA to a neighbor.

See the “Configuring Networks in OSPFv2” section on page 4-16 for information about the hello interval and dead timer.

BEFORE YOU BEGIN

Ensure that you have enabled the OSPF feature (see the “Enabling the OSPFv2 Feature” section on page 4-13).

SUMMARY STEPS

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **timers lsa-arrival** *msec*
4. **timers lsa-group-pacing** *seconds*
5. **timers throttle lsa** *start-time hold-interval max-time*
6. **timers throttle spf** *delay-time hold-time max-time*
7. **interface** *type slot/port*
8. **no switchport**
9. **ip ospf hello-interval** *seconds*
10. **ip ospf dead-interval** *seconds*
11. **ip ospf retransmit-interval** *seconds*
12. **ip ospf transmit-delay** *seconds*
13. (Optional) **show ip ospf**
14. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf <i>instance-tag</i> Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	timers lsa-arrival <i>msec</i> Example: switch(config-router)# timers lsa-arrival 2000	Sets the LSA arrival time in milliseconds. The range is from 10 to 600000. The default is 1000 milliseconds.

	Command	Purpose
Step 4	timers lsa-group-pacing <i>seconds</i> Example: switch(config-router)# timers lsa-group-pacing 1800	Sets the interval in seconds for grouping LSAs. The range is from 1 to 1800. The default is 240 seconds.
Step 5	timers throttle lsa <i>start-time</i> <i>hold-interval</i> <i>max-time</i> Example: switch(config-router)# timers throttle lsa 3000 6000 6000	Sets the rate limit in milliseconds for generating LSAs with the following timers: <i>start-time</i> —The range is from 50 to 5000 milliseconds. The default value is 50 milliseconds. <i>hold-interval</i> —The range is from 50 to 30,000 milliseconds. The default value is 5000 milliseconds. <i>max-time</i> —The range is from 50 to 30,000 milliseconds. The default value is 5000 milliseconds.
Step 6	timers throttle spf <i>delay-time</i> <i>hold-time</i> <i>max-time</i> Example: switch(config-router)# timers throttle spf 3000 2000 5000	Sets the SPF best-path schedule initial delay time, minimum hold time, and maximum wait time in milliseconds between SPF best-path calculations. The range is from 1 to 600000 milliseconds. The default values are a 200-ms delay time, 1000-ms hold time, and 5000-ms wait time.
Step 7	interface <i>type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 8	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 9	ip ospf hello-interval <i>seconds</i> Example: switch(config-if)# ip ospf retransmit-interval 30	Sets the hello interval this interface. The range is from 1 to 65535. The default is 10.
Step 10	ip ospf dead-interval <i>seconds</i> Example: switch(config-if)# ip ospf dead-interval 30	Sets the dead interval for this interface. The range is from 1 to 65535.
Step 11	ip ospf retransmit-interval <i>seconds</i> Example: switch(config-if)# ip ospf retransmit-interval 30	Sets the estimated time in seconds between LSAs transmitted from this interface. The range is from 1 to 65535. The default is 5.
Step 12	ip ospf transmit-delay <i>seconds</i> Example: switch(config-if)# ip ospf transmit-delay 450 switch(config-if)#	Sets the estimated time in seconds to transmit an LSA to a neighbor. The range is from 1 to 450. The default is 1.

	Command	Purpose
Step 13	show ip ospf Example: switch(config-if)# show ip ospf	(Optional) Displays information about OSPF.
Step 14	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to control LSA flooding with the lsa-group-pacing option:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# timers lsa-group-pacing 300
switch(config-router)# copy running-config startup-config
```

Restarting an OSPFv2 Instance

You can restart an OSPFv2 instance. This clears all neighbors for the instance.

To restart an OSPFv2 instance and remove all associated neighbors, use the following command:

Command	Purpose
restart ospf instance-tag Example: switch(config)# restart ospf 201	Restarts the OSPFv2 instance and removes all neighbors.

Configuring OSPFv2 with Virtualization

You can create multiple VRFs and use the same or multiple OSPFv2 instances in each VRF. You assign an OSPFv2 interface to a VRF.



Note

Configure all other parameters for an interface after you configure the VRF for an interface. Configuring a VRF for an interface deletes all the configuration for that interface.

BEFORE YOU BEGIN

Ensure that you have enabled the OSPF feature (see the [“Enabling the OSPFv2 Feature”](#) section on page 4-13).

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **router ospf** *instance-tag*

4. **vrf** *vrf-name*
5. **maximum-paths** *paths*
6. **interface** *interface-type slot/port*
7. **no switchport**
8. **vrf member** *vrf-name*
9. **ip-address** *ip-prefix/length*
10. **ip router ospf** *instance-tag area area-id*
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	vrf context <i>vrf-name</i> Example: switch(config)# vrf context RemoteOfficeVRF switch(config-vrf)#	Creates a new VRF and enters VRF configuration mode.
Step 3	router ospf <i>instance-tag</i> Example: switch(config-vrf)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 4	vrf <i>vrf-name</i> Example: switch(config-router)# vrf RemoteOfficeVRF switch(config-router-vrf)#	Enters VRF configuration mode.
Step 5	maximum-paths <i>paths</i> Example: switch(config-router-vrf)# maximum-paths 4	(Optional) Configures the maximum number of equal OSPFv2 paths to a destination in the route table for this VRF. Used for load balancing.
Step 6	interface <i>interface-type slot/port</i> Example: switch(config-router-vrf)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 7	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.

	Command	Purpose
Step 8	vrf member <i>vrf-name</i> Example: switch(config-if)# vrf member RemoteOfficeVRF	Adds this interface to a VRF.
Step 9	ip address <i>ip-prefix/length</i> Example: switch(config-if)# ip address 192.0.2.1/16	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 10	ip router ospf <i>instance-tag area area-id</i> Example: switch(config-if)# ip router ospf 201 area 0	Assigns this interface to the OSPFv2 instance and area configured.
Step 11	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to create a VRF and add an interface to the VRF:

```
switch# configure terminal
switch(config)# vrf context NewVRF
switch(config)# router ospf 201
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# vrf member NewVRF
switch(config-if)# ip address 192.0.2.1/16
switch(config-if)# ip router ospf 201 area 0
switch(config)# copy running-config startup-config
```

Verifying the OSPFv2 Configuration

To display the OSPFv2 configuration information, perform one of the following tasks:

Command	Purpose
show ip ospf	Displays the OSPFv2 configuration.
show ip ospf border-routers [vrf { <i>vrf-name</i> all default management }]	Displays the OSPFv2 border router configuration.
show ip ospf database [vrf { <i>vrf-name</i> all default management }]	Displays the OSPFv2 link-state database summary.
show ip ospf interface <i>number</i> [vrf { <i>vrf-name</i> all default management }]	Displays the OSPFv2 interface configuration.
show ip ospf lsa-content-changed-list <i>interface-type number</i>	Displays the OSPFv2 LSAs that have changed.

Command	Purpose
show ip ospf neighbors [<i>neighbor-id</i>] [detail] [<i>interface-type number</i>] [vrf { <i>vrf-name</i> all default management }] [summary]	Displays the list of OSPFv2 neighbors.
show ip ospf request-list <i>neighbor-id</i> [<i>interface-type number</i>]	Displays the list of OSPFv2 link-state requests.
show ip ospf retransmission-list <i>neighbor-id</i> [<i>interface-type number</i>]	Displays the list of OSPFv2 link-state retransmissions.
show ip ospf route [<i>ospf-route</i>] [summary] [vrf { <i>vrf-name</i> all default management }]	Displays the internal OSPFv2 routes.
show ip ospf summary-address [vrf { <i>vrf-name</i> all default management }]	Displays information about the OSPFv2 summary addresses.
show ip ospf virtual-links [brief] [vrf { <i>vrf-name</i> all default management }]	Displays information about OSPFv2 virtual links.
show ip ospf vrf { <i>vrf-name</i> all default management }	Displays information about the VRF-based OSPFv2 configuration.
show running-configuration ospf	Displays the current running OSPFv2 configuration.

Displaying OSPFv2 Statistics

To display OSPFv2 statistics, use the following commands:

Command	Purpose
show ip ospf policy statistics area <i>area-id</i> filter-list { in out } [vrf { <i>vrf-name</i> all default management }]	Displays the OSPFv2 route policy statistics for an area.
show ip ospf policy statistics redistribute { bgp <i>id</i> direct eigrp <i>id</i> ospf <i>id</i> rip <i>id</i> static } [vrf { <i>vrf-name</i> all default management }]	Displays the OSPFv2 route policy statistics.
show ip ospf statistics [vrf { <i>vrf-name</i> all default management }]	Displays the OSPFv2 event counters.
show ip ospf traffic [<i>interface-type</i> <i>number</i>] [vrf { <i>vrf-name</i> all default management }]	Displays the OSPFv2 packet counters.

Configuration Examples for OSPFv2

This example shows how to configure OSPFv2:

```
feature ospf
router ospf 201
router-id 290.0.2.1
```

```
interface ethernet 1/2
no switchport
ip router ospf 201 area 0.0.0.10
ip ospf authentication
ip ospf authentication-key 0 mypass
```

Additional References

For additional information related to implementing OSPF, see the following sections:

- [Related Documents, page 4-43](#)
- [MIBs, page 4-43](#)

Related Documents

Related Topic	Document Title
OSPFv2 CLI commands	<i>Cisco Nexus 3000 Series Command Reference</i>
Route maps	Chapter 16, “Configuring Route Policy Manager”

MIBs

MIBs	MIBs Link
<ul style="list-style-type: none">• OSPF-MIB• OSPF-TRAP-MIB	To locate and download MIBs, go to the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml



Configuring OSPFv3

This chapter describes how to configure Open Shortest Path First version 3 (OSPFv3) for IPv6 networks on the Cisco NX-OS device.

This chapter includes the following sections:

- [Information About OSPFv3, page 5-1](#)
- [Licensing Requirements for OSPFv3, page 5-12](#)
- [Prerequisites for OSPFv3, page 5-13](#)
- [Guidelines and Limitations for OSPFv3, page 5-13](#)
- [Default Settings, page 5-13](#)
- [Configuring Basic OSPFv3, page 5-14](#)
- [Configuring Advanced OSPFv3, page 5-20](#)
- [Verifying the OSPFv3 Configuration, page 5-40](#)
- [Monitoring OSPFv3, page 5-41](#)
- [Configuration Examples for OSPFv3, page 5-41](#)
- [Related Topics, page 5-41](#)
- [Additional References, page 5-41](#)

Information About OSPFv3

OSPFv3 is an IETF link-state protocol (see “[Overview](#)” section on page 1-1). An OSPFv3 router sends a special message, called a *hello packet*, out each OSPF-enabled interface to discover other OSPFv3 neighbor routers. Once a neighbor is discovered, the two routers compare information in the Hello packet to determine if the routers have compatible configurations. The neighbor routers attempt to establish *adjacency*, which means that the routers synchronize their link-state databases to ensure that they have identical OSPFv3 routing information. Adjacent routers share *link-state advertisements* (LSAs) that include information about the operational state of each link, the cost of the link, and any other neighbor information. The routers then flood these received LSAs out every OSPF-enabled interface so that all OSPFv3 routers eventually have identical link-state databases. When all OSPFv3 routers have identical link-state databases, the network is *converged* (see the “[Convergence](#)” section on page 1-6). Each router then uses Dijkstra’s Shortest Path First (SPF) algorithm to build its route table.

You can divide OSPFv3 networks into areas. Routers send most LSAs only within one area, which reduces the CPU and memory requirements for an OSPF-enabled router.

OSPFv3 supports IPv6. For information about OSPF for IPv4, see [Chapter 4, “Configuring OSPFv2”](#).

This section includes the following topics:

- [Comparison of OSPFv3 and OSPFv2, page 5-2](#)
- [Hello Packet, page 5-2](#)
- [Neighbors, page 5-3](#)
- [Adjacency, page 5-3](#)
- [Designated Routers, page 5-4](#)
- [Areas, page 5-5](#)
- [Link-State Advertisement, page 5-5](#)
- [OSPFv3 and the IPv6 Unicast RIB, page 5-8](#)
- [Address Family Support, page 5-8](#)
- [Advanced Features, page 5-8](#)

Comparison of OSPFv3 and OSPFv2

Much of the OSPFv3 protocol is the same as in OSPFv2. OSPFv3 is described in RFC 2740.

The key differences between the OSPFv3 and OSPFv2 protocols are as follows:

- OSPFv3 expands on OSPFv2 to provide support for IPv6 routing prefixes and the larger size of IPv6 addresses.
- LSAs in OSPFv3 are expressed as prefix and prefix length instead of address and mask.
- The router ID and area ID are 32-bit numbers with no relationship to IPv6 addresses.
- OSPFv3 uses link-local IPv6 addresses for neighbor discovery and other features.
- OSPFv3 uses IPv6 for authentication.
- OSPFv3 redefines LSA types.

Hello Packet

OSPFv3 routers periodically send Hello packets on every OSPF-enabled interface. The *hello interval* determines how frequently the router sends these Hello packets and is configured per interface. OSPFv3 uses Hello packets for the following tasks:

- Neighbor discovery
- Keepalives
- Bidirectional communications
- Designated router election (see the [“Designated Routers” section on page 5-4](#))

The Hello packet contains information about the originating OSPFv3 interface and router, including the assigned OSPFv3 cost of the link, the hello interval, and optional capabilities of the originating router. An OSPFv3 interface that receives these Hello packets determines if the settings are compatible with the receiving interface settings. Compatible interfaces are considered neighbors and are added to the neighbor table (see the [“Neighbors” section on page 5-3](#)).

Hello packets also include a list of router IDs for the routers that the originating interface has communicated with. If the receiving interface sees its own router ID in this list, then bidirectional communication has been established between the two interfaces.

OSPFv3 uses Hello packets as a keepalive message to determine if a neighbor is still communicating. If a router does not receive a Hello packet by the configured *dead interval* (usually a multiple of the hello interval), then the neighbor is removed from the local neighbor table.

Neighbors

An OSPFv3 interface must have a compatible configuration with a remote interface before the two can be considered neighbors. The two OSPFv3 interfaces must match the following criteria:

- Hello interval
- Dead interval
- Area ID (see the “[Areas](#)” section on page 5-5)
- Authentication
- Optional capabilities

If there is a match, the information is entered into the neighbor table:

- Neighbor ID—The router ID of the neighbor router.
- Priority—Priority of the neighbor router. The priority is used for designated router election (see the “[Designated Routers](#)” section on page 5-4).
- State—Indication of whether the neighbor has just been heard from, is in the process of setting up bidirectional communications, is sharing the link-state information, or has achieved full adjacency.
- Dead time—Indication of how long since the last Hello packet was received from this neighbor.
- Link-local IPv6 Address—The link-local IPv6 address of the neighbor.
- Designated Router—Indication of whether the neighbor has been declared the designated router or backup designated router (see the “[Designated Routers](#)” section on page 5-4).
- Local interface—The local interface that received the Hello packet for this neighbor.

When the first Hello packet is received from a new neighbor, the neighbor is entered into the neighbor table in the initialization state. Once bidirectional communication is established, the neighbor state becomes two-way. ExStart and exchange states come next, as the two interfaces exchange their link-state database. Once this is all complete, the neighbor moves into the full state, which signifies full adjacency. If the neighbor fails to send any Hello packets in the dead interval, then the neighbor is moved to the down state and is no longer considered adjacent.

Adjacency

Not all neighbors establish adjacency. Depending on the network type and designated router establishment, some neighbors become fully adjacent and share LSAs with all their neighbors, while other neighbors do not. For more information, see the “[Designated Routers](#)” section on page 5-4.

Adjacency is established using Database Description packets, Link State Request packets, and Link State Update packets in OSPFv3. The Database Description packet includes the LSA headers from the link-state database of the neighbor (see the “[Link-State Database](#)” section on page 5-7). The local router compares these headers with its own link-state database and determines which LSAs are new or updated.

The local router sends a Link State Request packet for each LSA that it needs new or updated information on. The neighbor responds with a Link State Update packet. This exchange continues until both routers have the same link-state information.

Designated Routers

Networks with multiple routers present a unique situation for OSPFv3. If every router floods the network with LSAs, the same link-state information is sent from multiple sources. Depending on the type of network, OSPFv3 might use a single router, the *designated router (DR)*, to control the LSA floods and represent the network to the rest of the OSPFv3 area (see the “Areas” section on page 5-5). If the DR fails, OSPFv3 selects a *backup designated router (BDR)*. If the DR fails, OSPFv3 uses the BDR.

Network types are as follows:

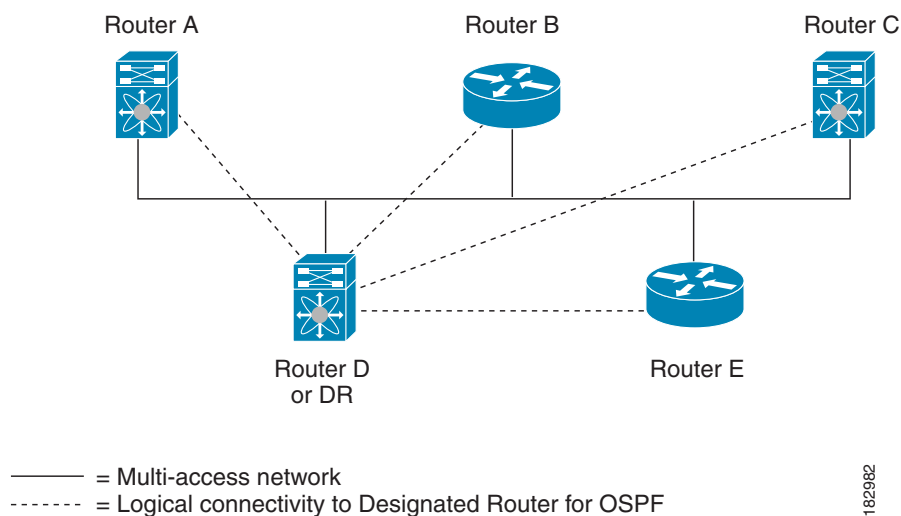
- **Point-to-point**—A network that exists only between two routers. All neighbors on a point-to-point network establish adjacency and there is no DR.
- **Broadcast**—A network with multiple routers that can communicate over a shared medium that allows broadcast traffic, such as Ethernet. OSPFv3 routers establish a DR and BDR that controls LSA flooding on the network. OSPFv3 uses a MAC address of 0100.5300.0005 to communicate with neighbors.

The DR and BDR are selected based on the information in the Hello packet. When an interface sends a Hello packet, it sets the priority field and the DR and BDR field if it knows who the DR and BDR are. The routers follow an election procedure based on which routers declare themselves in the DR and BDR fields and the priority field in the Hello packet. As a final determinant, OSPFv3 chooses the highest router IDs as the DR and BDR.

All other routers establish adjacency with the DR and the BDR. Figure 5-1 shows this adjacency relationship between all routers and the DR.

DRs are based on a router interface. A router might be the DR for one network and not for another network on a different interface.

Figure 5-1 DR in Multi-Access Network



182982

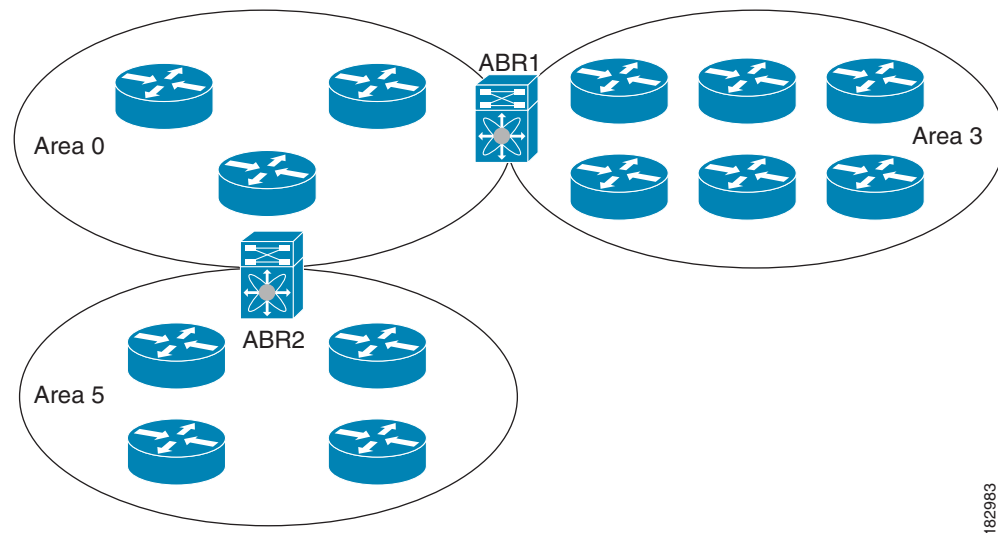
Areas

You can limit the CPU and memory requirements that OSPFv3 puts on the routers by dividing an OSPFv3 network into *areas*. An area is a logical division of routers and links within an OSPFv3 domain that creates separate subdomains. LSA flooding is contained within an area, and the link-state database is limited to links within the area. You can assign an area ID to the interfaces within the defined area. The Area ID is a 32-bit value that can be expressed as a number or in dotted decimal notation, such as 10.2.3.1.

Cisco NX-OS always displays the area in dotted decimal notation.

If you define more than one area in an OSPFv3 network, you must also define the backbone area, which has the reserved area ID of 0. If you have more than one area, then one or more routers become area border routers (ABRs). An ABR connects to both the backbone area and at least one other defined area (see [Figure 5-2](#)).

Figure 5-2 OSPFv3 Areas



182983

The ABR has a separate link-state database for each area which it connects to. The ABR sends Inter-Area Prefix (type 3) LSAs (see the [“Route Summarization”](#) section on page 5-10) from one connected area to the backbone area. The backbone area sends summarized information about one area to another area. In [Figure 5-2](#), Area 0 sends summarized information about Area 5 to Area 3.

OSPFv3 defines one other router type: the autonomous system boundary router (ASBR). This router connects an OSPFv3 area to another autonomous system. An autonomous system is a network controlled by a single technical administration entity. OSPFv3 can redistribute its routing information into another autonomous system or receive redistributed routes from another autonomous system. For more information, see the [“Advanced Features”](#) section on page 5-8.

Link-State Advertisement

OSPFv3 uses link-state advertisements (LSAs) to build its routing table.

This section includes the following topics:

- [LSA Types, page 5-6](#)
- [Link Cost, page 5-6](#)
- [Flooding and LSA Group Pacing, page 5-7](#)
- [Link-State Database, page 5-7](#)

LSA Types

[Table 5-1](#) shows the LSA types supported by Cisco NX-OS.

Table 5-1 LSA Types

Type	Name	Description
1	Router LSA	LSA sent by every router. This LSA includes the state and cost of all links but does not include prefix information. Router LSAs trigger an SPF recalculation. Router LSAs are flooded to the local OSPFv3 area.
2	Network LSA	LSA sent by the DR. This LSA lists all routers in the multi-access network but does not include prefix information. Network LSAs trigger an SPF recalculation. See the “Designated Routers” section on page 5-4 .
3	Inter-Area Prefix LSA	LSA sent by the area border router to an external area for each destination in local area. This LSA includes the link cost from the border router to the local destination. See the “Areas” section on page 5-5 .
4	Inter-Area Router LSA	LSA sent by the area border router to an external area. This LSA advertises the link cost to the ASBR only. See the “Areas” section on page 5-5 .
5	AS External LSA	LSA generated by the ASBR. This LSA includes the link cost to an external autonomous system destination. AS External LSAs are flooded throughout the autonomous system. See the “Areas” section on page 5-5 .
7	Type-7 LSA	LSA generated by the ASBR within an NSSA. This LSA includes the link cost to an external autonomous system destination. Type-7 LSAs are flooded only within the local NSSA. See the “Areas” section on page 5-5 .
8	Link LSA	LSA sent by every router, using a link-local flooding scope (see the “Flooding and LSA Group Pacing” section on page 5-7). This LSA includes the link-local address and IPv6 prefixes for this link.
9	Intra-Area Prefix LSA	LSA sent by every router. This LSA includes any prefix or link state changes. Intra-Area Prefix LSAs are flooded to the local OSPFv3 area. This LSA does not trigger an SPF recalculation.
11	Grace LSAs	LSA sent by a restarting router, using a link-local flooding scope. This LSA is used for a graceful restart of OSPFv3. See the “High Availability and Graceful Restart” section on page 5-11 .

Link Cost

Each OSPFv3 interface is assigned a [link cost](#). The cost is an arbitrary number. By default, Cisco NX-OS assigns a cost that is the configured reference bandwidth divided by the interface bandwidth. By default, the reference bandwidth is 40 Gb/s. The link cost is carried in the LSA updates for each link.

Flooding and LSA Group Pacing

OSPFv3 floods LSA updates to different sections of the network, depending on the LSA type. OSPFv3 uses the following flooding scopes:

- Link-local—LSA is flooded only on the local link. Used for Link LSAs and Grace LSAs.
- Area-local—LSA is flooded throughout a single OSPF area only. Used for Router LSAs, Network LSAs, Inter-Area-Prefix LSAs, Inter-Area-Router LSAs, and Intra-Area-Prefix LSAs.
- AS scope—LSA is flooded throughout the routing domain. An AS scope is used for AS External LSAs.

LSA flooding guarantees that all routers in the network have identical routing information. LSA flooding depends on the OSPFv3 area configuration (see the [“Areas” section on page 5-5](#)). The LSAs are flooded based on the *link-state refresh* time (every 30 minutes by default). Each LSA has its own link-state refresh time.

You can control the flooding rate of LSA updates in your network by using the LSA group pacing feature. LSA group pacing can reduce high CPU or buffer utilization. This feature groups LSAs with similar link-state refresh times to allow OSPFv3 to pack multiple LSAs into an OSPFv3 Update message.

By default, LSAs with link-state refresh times within 10 seconds of each other are grouped together. You should lower this value for large link-state databases or raise it for smaller databases to optimize the OSPFv3 load on your network.

Link-State Database

Each router maintains a link-state database for the OSPFv3 network. This database contains all the collected LSAs and includes information on all the routes through the network. OSPFv3 uses this information to calculate the best path to each destination and populates the routing table with these best paths.

LSAs are removed from the link-state database if no LSA update has been received within a set interval, called the MaxAge. Routers flood a repeat of the LSA every 30 minutes to prevent accurate link-state information from being aged out. Cisco NX-OS supports the LSA grouping feature to prevent all LSAs from refreshing at the same time. For more information, see the [“Flooding and LSA Group Pacing” section on page 5-7](#).

Multi-Area Adjacency

OSPFv3 multi-area adjacency allows you to configure a link on the primary interface that is in more than one area. This link becomes the preferred intra-area link in those areas. Multi-area adjacency establishes a point-to-point unnumbered link in an OSPFv3 area that provides a topological path for that area. The primary adjacency uses the link to advertise an unnumbered point-to-point link in the Router LSA for the corresponding area when the neighbor state is full.

The multi-area interface exists as a logical construct over an existing primary interface for OSPF; however, the neighbor state on the primary interface is independent of the multi-area interface. The multi-area interface establishes a neighbor relationship with the corresponding multi-area interface on the neighboring router. See the [“Configuring Multi-Area Adjacency” section on page 5-25](#) for more information.

OSPFv3 and the IPv6 Unicast RIB

OSPFv3 runs the Dijkstra shortest path first algorithm on the link-state database. This algorithm selects the best path to each destination based on the sum of all the link costs for each link in the path. The shortest path for each destination is then put in the OSPFv3 route table. When the OSPFv3 network is converged, this route table feeds into the IPv6 unicast RIB. OSPFv3 communicates with the IPv6 unicast RIB to do the following:

- Add or remove routes
- Handle route redistribution from other protocols
- Provide convergence updates to remove stale OSPFv3 routes and for stub router advertisements (see the [“Multiple OSPFv3 Instances”](#) section on page 5-12)

OSPFv3 also runs a modified Dijkstra algorithm for fast recalculation for Inter-Area Prefix, Inter-Area Router, AS-External, type-7, and Intra-Area Prefix (type 3, 4, 5, 7, 8) LSA changes.

Address Family Support

Cisco NX-OS supports multiple address families, such as unicast IPv6. OSPFv3 features that are specific to an *address family* are as follows:

- Default routes
- Route summarization
- Route redistribution
- Filter lists for border routers
- SPF optimization

Use the **address-family ipv6 unicast** command to enter the IPv6 unicast address family configuration mode when configuring these features.

Advanced Features

Cisco NX-OS supports advanced OSPFv3 features that enhance the usability and scalability of OSPFv3 in the network.

This section includes the following topics:

- [Stub Area, page 5-9](#)
- [Not-So-Stubby Area, page 5-9](#)
- [Virtual Links, page 5-10](#)
- [Route Redistribution, page 5-10](#)
- [Route Summarization, page 5-10](#)
- [High Availability and Graceful Restart, page 5-11](#)
- [Multiple OSPFv3 Instances, page 5-12](#)
- [SPF Optimization, page 5-12](#)
- [BFD, page 5-12](#)
- [Virtualization Support, page 5-12](#)

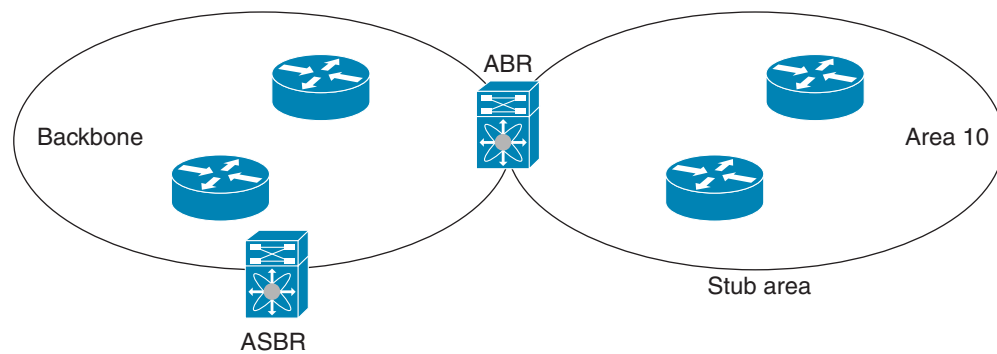
Stub Area

You can limit the amount of external routing information that floods an area by making it a *stub area*. A stub area is an area that does not allow AS External (type 5) LSAs (see the [“Link-State Advertisement” section on page 5-5](#)). These LSAs are usually flooded throughout the local autonomous system to propagate external route information. Stub areas have the following requirements:

- All routers in the stub area are stub routers. See the [“Stub Routing” section on page 1-7](#).
- No ASBR routers exist in the stub area.
- You cannot configure virtual links in the stub area.

Figure 5-3 shows an example an OSPFv3 autonomous system where all routers in area 0.0.0.10 have to go through the ABR to reach external autonomous systems. Area 0.0.0.10 can be configured as a stub area.

Figure 5-3 Stub Area



Stub areas use a default route for all traffic that needs to go through the backbone area to the external autonomous system. The default route is an Inter-Area-Prefix LSA with the prefix length set to 0 for IPv6.

Not-So-Stubby Area

A Not-So-Stubby Area (*NSSA*) is similar to the stub area, except that an NSSA allows you to import autonomous system external routes within an NSSA using redistribution. The NSSA ASBR redistributes these routes and generates type-7 LSAs that it floods throughout the NSSA. You can optionally configure the ABR that connects the NSSA to other areas to translate this type-7 LSA to AS External (type 5) LSAs. The ABR then floods these AS External LSAs throughout the OSPFv3 autonomous system. Summarization and filtering are supported during the translation. See the [“Link-State Advertisement” section on page 5-5](#) for details on type-7 LSAs.

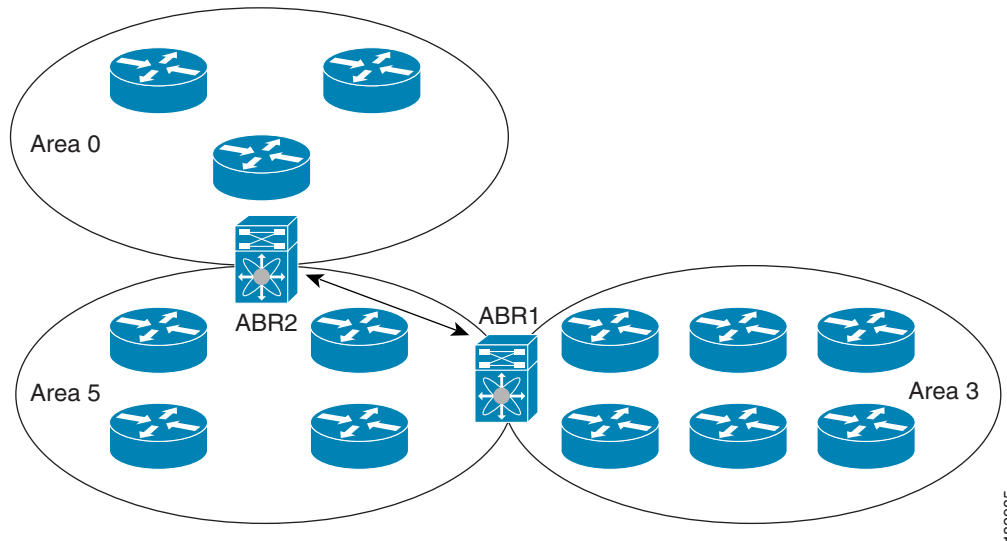
You can, for example, use NSSA to simplify administration if you are connecting a central site using OSPFv3 to a remote site that is using a different routing protocol. Before NSSA, the connection between the corporate site border router and a remote router could not be run as an OSPFv3 stub area because routes for the remote site could not be redistributed into a stub area. With NSSA, you can extend OSPFv3 to cover the remote connection by defining the area between the corporate router and remote router as an NSSA (see the [“Configuring NSSA” section on page 5-23](#)).

The backbone Area 0 cannot be an NSSA.

Virtual Links

Virtual links allow you to connect an OSPFv3 area ABR to a backbone area ABR when a direct physical connection is not available. [Figure 5-4](#) shows a virtual link that connects Area 3 to the backbone area through Area 5.

Figure 5-4 Virtual Links



You can also use virtual links to temporarily recover from a partitioned area, which occurs when a link within the area fails, isolating part of the area from reaching the designated ABR to the backbone area.

Route Redistribution

OSPFv3 can learn routes from other routing protocols by using route redistribution. See the [“Route Redistribution” section on page 1-6](#). You configure OSPFv3 to assign a link cost for these redistributed routes or a default link cost for all redistributed routes.

Route redistribution uses route maps to control which external routes are redistributed. You must configure a route map with the redistribution to control which routes are passed into OSPFv2. A route map allows you to filter routes based on attributes such as the destination, origination protocol, route type, route tag, and so on. You can use route maps to modify parameters in the AS External (type 5) and NSSA External (type 7) LSAs before these external routes are advertised in the local OSPFv3 autonomous system. For more information, see [Chapter 16, “Configuring Route Policy Manager,”](#)

Route Summarization

Because OSPFv3 shares all learned routes with every OSPF-enabled router, you might want to use route summarization to reduce the number of unique routes that are flooded to every OSPF-enabled router. Route summarization simplifies route tables by replacing more-specific addresses with an address that represents all the specific addresses. For example, you can replace 2010:11:22:0:1000::1 and 2010:11:22:0:2000:679:1 with one summary address, 2010:11:22::/32.

Typically, you would summarize at the boundaries of area border routers (ABRs). Although you could configure summarization between any two areas, it is better to summarize in the direction of the backbone so that the backbone receives all the aggregate addresses and injects them, already summarized, into other areas. The two types of summarization are as follows:

- Inter-area route summarization
- External route summarization

You configure inter-area route summarization on ABRs, summarizing routes between areas in the autonomous system. To take advantage of summarization, assign network numbers in areas in a contiguous way to be able to lump these addresses into one range.

External route summarization is specific to external routes that are injected into OSPFv3 using route redistribution. You should make sure that external ranges that are being summarized are contiguous. Summarizing overlapping ranges from two different routers could cause packets to be sent to the wrong destination. Configure external route summarization on ASBRs that are redistributing routes into OSPF.

When you configure a summary address, Cisco NX-OS automatically configures a discard route for the summary address to prevent routing black holes and route loops.

High Availability and Graceful Restart

Cisco NX-OS supports high-availability. If a Cisco NX-OS system experiences a cold reboot, the network stops forwarding traffic to the system and removes the system from the network topology. In this scenario, OSPFv3 experiences a stateless restart, and removes all neighbor adjacencies on the local system. Cisco NX-OS applies the startup configuration and OSPFv3 rediscovers the neighbors and establishes the adjacencies again.

OSPFv3 automatically restarts if the process experiences problems. After the restart, OSPFv3 initiates a graceful restart so that the platform is not taken out of the network topology. If you manually restart OSPF, it performs a graceful restart, which is similar to a stateful switchover. The running configuration is applied in both cases.

A graceful restart, or nonstop forwarding (NSF), allows OSPFv3 to remain in the data forwarding path through a process restart. When OSPFv3 needs to restart, it first sends a link-local Grace (type 11) LSA. This restarting OSPFv3 platform is called NSF capable.

The Grace LSA includes a grace period, which is a specified time that the neighbor OSPFv3 interfaces hold onto the LSAs from the restarting OSPFv3 interface. (Typically, OSPFv3 tears down the adjacency and discards all LSAs from a down or restarting OSPFv3 interface.) The participating neighbors, which are called NSF helpers, keep all LSAs that originate from the restarting OSPFv3 interface as if the interface were still adjacent.

When the restarting OSPFv3 interface is operational again, it rediscovers its neighbors, establishes adjacency, and starts sending its LSA updates again. At this point, the NSF helpers recognize that graceful restart has finished.



Note

If the restarting OSPFv3 interface does not come back up before the end of the grace period, or if the network experiences a topology change, the OSPFv3 neighbors tear down adjacency with the restarting OSPFv3 and treat it as a normal OSPFv3 restart.

**Note**

You must enable graceful restart to support an in-service software upgrade (ISSU) for OSPFv3. If you disable graceful restart, Cisco NX-OS issues a warning that ISSU cannot be supported with this configuration.

Multiple OSPFv3 Instances

Cisco NX-OS supports multiple instances of the OSPFv3 protocol. By default, every instance uses the same system router ID. You must manually configure the router ID for each instance if the instances are in the same OSPFv3 autonomous system.

The OSPFv3 header includes an instance ID field to identify that OSPFv3 packet for a particular OSPFv3 instance. You can assign the OSPFv3 instance. The interface drops all OSPFv3 packets that do not have a matching OSPFv3 instance ID in the packet header.

Cisco NX-OS allows only one OSPFv3 instance on an interface.

SPF Optimization

Cisco NX-OS optimizes the SPF algorithm in the following ways:

- Partial SPF for Network (type 2) LSAs, Inter-Area Prefix (type 3) LSAs, and AS External (type 5) LSAs—When there is a change on any of these LSAs, Cisco NX-OS performs a faster partial calculation rather than running the whole SPF calculation.
- SPF timers—You can configure different timers for controlling SPF calculations. These timers include exponential backoff for subsequent SPF calculations. The exponential backoff limits the CPU load of multiple SPF calculations.

BFD

This feature supports bidirectional forwarding detection (BFD). BFD is a detection protocol designed to provide fast forwarding-path failure detection times. 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.

Virtualization Support

OSPFv3 supports virtual routing and forwarding (VRF) instances.

Licensing Requirements for OSPFv3

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	OSPFv3 requires an Enterprise Services license. For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the <i>Cisco NX-OS Licensing Guide</i> .

Prerequisites for OSPFv3

OSPFv3 has the following prerequisites:

- You must be familiar with routing fundamentals to configure OSPFv3.
- You must be logged on to the switch.
- You have configured at least one interface for IPv6 that is capable of communicating with a remote OSPFv3 neighbor.
- You have installed the Enterprise Services license.
- You have completed the OSPFv3 network strategy and planning for your network. For example, you must decide whether multiple areas are required.
- You have enabled OSPF (see the [“Enabling OSPFv3” section on page 5-14](#)).
- You have installed the Advanced Services license.
- You are familiar with IPv6 addressing and basic configuration. See [Chapter 3, “Configuring IPv6”](#) for information on IPv6 routing and addressing.

Guidelines and Limitations for OSPFv3

OSPFv3 has the following configuration guidelines and limitations:

- You can have up to four instances of OSPFv3 in a VDC.
- Cisco NX-OS displays areas in dotted decimal notation regardless of whether you enter the area in decimal or dotted decimal notation.
- If you configure OSPFv3 in a virtual port channel (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
```

**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.

Default Settings

[Table 5-2](#) lists the default settings for OSPFv3 parameters.

Table 5-2 **Default OSPFv3 Parameters**

Parameters	Default
Hello interval	10 seconds
Dead interval	40 seconds
Graceful restart grace period	60 seconds
Graceful restart notify period	15 seconds
OSPFv3 feature	Disabled

Table 5-2 *Default OSPFv3 Parameters (continued)*

Parameters	Default
Stub router advertisement announce time	600 seconds
Reference bandwidth for link cost calculation	40 Gb/s
LSA minimal arrival time	1000 milliseconds
LSA group pacing	10 seconds
SPF calculation initial delay time	200 milliseconds
SPF calculation minimum hold time	1000 milliseconds
SPF calculation maximum wait time	5000 milliseconds

Configuring Basic OSPFv3

Configure OSPFv3 after you have designed your OSPFv3 network.

This section includes the following topics:

- [Enabling OSPFv3, page 5-14](#)
- [Creating an OSPFv3 Instance, page 5-15](#)
- [Configuring Networks in OSPFv3, page 5-17](#)

Enabling OSPFv3

You must enable OSPFv3 before you can configure OSPFv3.

SUMMARY STEPS

1. **configure terminal**
2. **feature ospfv3**
3. (Optional) **show feature**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	feature ospfv3 Example: switch(config)# feature ospfv3	Enables OSPFv3.

	Command	Purpose
Step 3	show feature	(Optional) Displays enabled and disabled features.
	Example: switch(config)# show feature	
Step 4	copy running-config startup-config	(Optional) Saves this configuration change.
	Example: switch(config)# copy running-config startup-config	

To disable the OSPFv3 feature and remove all associated configuration, use the following command in configuration mode.

Command	Purpose
no feature ospfv3	Disables the OSPFv3 feature and removes all associated configuration.
Example: switch(config)# no feature ospfv3	

Creating an OSPFv3 Instance

The first step in configuring OSPFv3 is to create an instance or OSPFv3 instance. You assign a unique instance tag for this OSPFv3 instance. The instance tag can be any string. For each OSPFv3 instance, you can also configure the following optional parameters:

- Router ID—Configures the router ID for this OSPFv3 instance. If you do not use this parameter, the router ID selection algorithm is used. For more information, see the [“Router IDs” section on page 1-5](#).
- Administrative distance—Rates the trustworthiness of a routing information source. For more information, see the [“Administrative Distance” section on page 1-7](#).
- Log adjacency changes—Creates a system message whenever an OSPFv3 neighbor changes its state.
- Maximum paths—Sets the maximum number of equal paths that OSPFv3 installs in the route table for a particular destination. Use this parameter for load balancing between multiple paths.
- Reference bandwidth—Controls the calculated OSPFv3 cost metric for a network. The calculated cost is the reference bandwidth divided by the interface bandwidth. You can override the calculated cost by assigning a link cost when a network is added to the OSPFv3 instance. For more information, see the [“Configuring Networks in OSPFv3” section on page 5-17](#).

For more information about OSPFv3 instance parameters, see the [“Configuring Advanced OSPFv3” section on page 5-20](#).

BEFORE YOU BEGIN

You must enable OSPFv3 (see the [“Enabling OSPFv3” section on page 5-14](#)).

Ensure that the OSPFv3 instance tag that you plan on using is not already in use on this router.

Use the **show ospfv3 instance-tag** command to verify that the instance tag is not in use.

OSPFv3 must be able to obtain a router identifier (for example, a configured loopback address) or you must configure the router ID option.

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 *instance-tag***
3. (Optional) **router-id *ip-address***
4. (Optional) **show ipv6 ospfv3 *instance-tag***
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospfv3 <i>instance-tag</i> Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	router-id <i>ip-address</i> Example: switch(config-router)# router-id 192.0.2.1	(Optional) Configures the OSPFv3 router ID. This ID uses the dotted decimal notation and identifies this OSPFv3 instance and must exist on a configured interface in the system.
Step 4	show ipv6 ospfv3 <i>instance-tag</i> Example: switch(config-router)# show ipv6 ospfv3 201	(Optional) Displays OSPFv3 information.
Step 5	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

To remove the OSPFv3 instance and all associated configuration, use the following command in configuration mode:

Command	Purpose
no router ospfv3 <i>instance-tag</i> Example: switch(config)# no router ospfv3 201	Deletes the OSPFv3 instance and all associated configuration.

**Note**

This command does not remove OSPF configuration in interface mode. You must manually remove any OSPFv3 commands configured in interface mode.

You can configure the following optional parameters for OSPFv3 in router configuration mode:

Command	Purpose
log-adjacency-changes [<i>detail</i>] Example: switch(config-router)# log-adjacency-changes	Generates a system message whenever a neighbor changes state.
passive-interface default Example: switch(config-router)# passive-interface default	Suppresses routing updates on all interfaces. This command is overridden by the VRF or interface command mode configuration.

You can configure the following optional parameters for OSPFv3 in address family configuration mode:

Command	Purpose
distance <i>number</i> Example: switch(config-router-af)# distance 25	Configures the administrative distance for this OSPFv3 instance. The range is from 1 to 255. The default is 110.
maximum-paths <i>paths</i> Example: switch(config-router-af)# maximum-paths 4	Configures the maximum number of equal OSPFv3 paths to a destination in the route table. The range is from 1 to 16. The default is 8. This command is used for load balancing.

This example shows how to create an OSPFv3 instance:

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# copy running-config startup-config
```

Configuring Networks in OSPFv3

You can configure a network to OSPFv3 by associating it through the interface that the router uses to connect to that network (see the [“Neighbors” section on page 5-3](#)). You can add all networks to the default backbone area (Area 0), or you can create new areas using any decimal number or an IP address.

**Note**

All areas must connect to the backbone area either directly or through a virtual link.

**Note**

OSPFv3 is not enabled on an interface until you configure a valid IPv6 address for that interface.

BEFORE YOU BEGIN

You must enable OSPFv3 (see the [“Enabling OSPFv3”](#) section on page 5-14).

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ipv6 address** *ipv6-prefix/length*
4. **ipv6 router ospfv3** *instance-tag area area-id* [**secondaries none**]
5. (Optional) **show ipv6 ospfv3** *instance-tag interface interface-type slot/port*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	ipv6 address <i>ipv6-prefix/length</i> Example: switch(config-if)# ipv6 address 2001:0DB8::1/48	Assigns an IPv6 address to this interface.
Step 4	ipv6 router ospfv3 <i>instance-tag area area-id</i> [secondaries none] Example: switch(config-if)# ipv6 router ospfv3 201 area 0	Adds the interface to the OSPFv3 instance and area.
Step 5	show ipv6 ospfv3 <i>instance-tag interface interface-type slot/port</i> Example: switch(config-if)# show ipv6 ospfv3 201 interface ethernet 1/2	(Optional) Displays OSPFv3 information.
Step 6	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

You can configure the following optional parameters for OSPFv3 in interface configuration mode:

Command	Purpose
ospfv3 cost <i>number</i> Example: switch(config-if)# ospfv3 cost 25	Configures the OSPFv3 cost metric for this interface. The default is to calculate a cost metric, based on the reference bandwidth and interface bandwidth. The range is from 1 to 65535.
ospfv3 dead-interval <i>seconds</i> Example: switch(config-if)# ospfv3 dead-interval 50	Configures the OSPFv3 dead interval, in seconds. The range is from 1 to 65535. The default is four times the hello interval, in seconds.
ospfv3 hello-interval <i>seconds</i> Example: switch(config-if)# ospfv3 hello-interval 25	Configures the OSPFv3 hello interval, in seconds. The range is from 1 to 65535. The default is 10 seconds.
ospfv3 instance <i>instance</i> Example: switch(config-if)# ospfv3 instance 25	Configures the OSPFv3 instance ID. The range is from 0 to 255. The default is 0. The instance ID is link-local in scope.
ospfv3 mtu-ignore Example: switch(config-if)# ospfv3 mtu-ignore	Configures OSPFv3 to ignore any IP maximum transmission unit (MTU) mismatch with a neighbor. The default is to not establish adjacency if the neighbor MTU does not match the local interface MTU.
ospfv3 network { broadcast point-point } Example: switch(config-if)# ospfv3 network broadcast	Sets the OSPFv3 network type.
[default no] ospfv3 passive-interface Example: switch(config-if)# ospfv3 passive-interface	Suppresses routing updates on the interface. This command overrides the router or VRF command mode configuration. The default option removes this interface mode command and reverts to the router or VRF configuration, if present.
ospfv3 priority <i>number</i> Example: switch(config-if)# ospfv3 priority 25	Configures the OSPFv3 priority, used to determine the DR for an area. The range is from 0 to 255. The default is 1. See the “Designated Routers” section on page 5-4 .
ospfv3 shutdown Example: switch(config-if)# ospfv3 shutdown	Shuts down the OSPFv3 instance on this interface.

This example shows how to add a network area 0.0.0.10 in OSPFv3 instance 201:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ipv6 address 2001:0DB8::1/48
switch(config-if)# ipv6 ospfv3 201 area 0.0.0.10
switch(config-if)# copy running-config startup-config
```

Configuring Advanced OSPFv3

Configure OSPFv3 after you have designed your OSPFv3 network.

This section includes the following topics:

- [Configuring Filter Lists for Border Routers, page 5-20](#)
- [Configuring Stub Areas, page 5-22](#)
- [Configuring a Totally Stubby Area, page 5-23](#)
- [Configuring NSSA, page 5-23](#)
- [Configuring Multi-Area Adjacency, page 5-25](#)
- [Configuring Virtual Links, page 5-27](#)
- [Configuring Redistribution, page 5-29](#)
- [Limiting the Number of Redistributed Routes, page 5-30](#)
- [Configuring Route Summarization, page 5-32](#)
- [Modifying the Default Timers, page 5-34](#)
- [Configuring Graceful Restart, page 5-36](#)
- [Restarting an OSPFv3 Instance, page 5-37](#)
- [Configuring OSPFv3 with Virtualization, page 5-38](#)

Configuring Filter Lists for Border Routers

You can separate your OSPFv3 domain into a series of areas that contain related networks. All areas must connect to the backbone area through an area border router (ABR). OSPFv3 domains can connect to external domains as well through an autonomous system border router (ASBR). See the [“Areas” section on page 5-5](#).

ABRs have the following optional configuration parameters:

- Area range—Configures route summarization between areas. For more information, see the [“Configuring Route Summarization” section on page 5-32](#).
- Filter list—Filters the Inter-Area Prefix (type 3) LSAs on an ABR that are allowed in from an external area.

ASBRs also support filter lists.

BEFORE YOU BEGIN

Create the route map that the filter list uses to filter IP prefixes in incoming or outgoing Inter-Area Prefix (type 3) LSAs. See [Chapter 16, “Configuring Route Policy Manager.”](#)

You must enable OSPFv3 (see the [“Enabling OSPFv3” section on page 5-14](#)).

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 *instance-tag***
3. **address-family ipv6 unicast**

4. **area** *area-id* **filter-list route-map** *map-name* {in | out}
5. (Optional) **show ipv6 ospfv3 policy statistics area** *id* **filter-list** {in | out}
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospfv3 <i>instance-tag</i> Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	address-family ipv6 unicast Example: switch(config-router)# address-family ipv6 unicast switch(config-router-af)#	Enters IPv6 unicast address family mode.
Step 4	area <i>area-id</i> filter-list route-map <i>map-name</i> {in out} Example: switch(config-router-af)# area 0.0.0.10 filter-list route-map FilterLSAs in	Filters incoming or outgoing Inter-Area Prefix (type 3) LSAs on an ABR.
Step 5	show ipv6 ospfv3 policy statistics area <i>id</i> filter-list {in out} Example: switch(config-if)# show ipv6 ospfv3 policy statistics area 0.0.0.10 filter-list in	(Optional) Displays OSPFv3 policy information.
Step 6	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to enable graceful restart if it has been disabled:

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# address-family ipv6 unicast
switch(config-router-af)# area 0.0.0.10 filter-list route-map FilterLSAs in
switch(config-router-af)# copy running-config startup-config
```

Configuring Stub Areas

You can configure a stub area for part of an OSPFv3 domain where external traffic is not necessary. Stub areas block AS External (type 5) LSAs, limiting unnecessary routing to and from selected networks. See the [“Stub Area” section on page 5-9](#). You can optionally block all summary routes from going into the stub area.

BEFORE YOU BEGIN

You must enable OSPF (see the [“Enabling OSPFv3” section on page 5-14](#)).

Ensure that there are no virtual links or ASBRs in the proposed stub area.

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 *instance-tag***
3. **area *area-id* stub**
4. (Optional) **address-family ipv6 unicast**
5. (Optional) **area *area-id* default-cost *cost***
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospfv3 <i>instance-tag</i> Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	area <i>area-id</i> stub Example: switch(config-router)# area 0.0.0.10 stub	Creates this area as a stub area.
Step 4	address-family ipv6 unicast Example: switch(config-router)# address-family ipv6 unicast switch(config-router-af)#	(Optional) Enters IPv6 unicast address family mode.

	Command	Purpose
Step 5	area <i>area-id</i> default-cost <i>cost</i> Example: switch(config-router-af)# area 0.0.0.10 default-cost 25	(Optional) Sets the cost metric for the default summary route sent into this stub area. The range is from 0 to 16777215.
Step 6	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to create a stub area that blocks all summary route updates:

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# area 0.0.0.10 stub no-summary
switch(config-router)# copy running-config startup-config
```

Configuring a Totally Stubby Area

You can create a totally stubby area and prevent all summary route updates from going into the stub area.

To create a totally stubby area, use the following command in router configuration mode:

Command	Purpose
area <i>area-id</i> stub no-summary Example: switch(config-router)# area 20 stub no-summary	Creates this area as a totally stubby area.

Configuring NSSA

You can configure an NSSA for part of an OSPFv3 domain where limited external traffic is required. See the [“Not-So-Stubby Area” section on page 5-9](#). You can optionally translate this external traffic to an AS External (type 5) LSA and flood the OSPFv3 domain with this routing information. An NSSA can be configured with the following optional parameters:

- **No redistribution**—Redistributes routes that bypass the NSSA to other areas in the OSPFv3 autonomous system. Use this option when the NSSA ASBR is also an ABR.
- **Default information originate**—Generates a Type-7 LSA for a default route to the external autonomous system. Use this option on an NSSA ASBR if the ASBR contains the default route in the routing table. This option can be used on an NSSA ABR whether or not the ABR contains the default route in the routing table.
- **Route map**—Filters the external routes so that only those routes you want are flooded throughout the NSSA and other areas.
- **Translate**—Translates Type-7 LSAs to AS External (type 5) LSAs for areas outside the NSSA. Use this command on an NSSA ABR to flood the redistributed routes throughout the OSPFv3 autonomous system. You can optionally suppress the forwarding address in these AS External LSAs.

- No summary—Blocks all summary routes from flooding the NSSA. Use this option on the NSSA ABR.

BEFORE YOU BEGIN

You must enable OSPF (see the [“Enabling OSPFv3” section on page 5-14](#)).

Ensure that there are no virtual links in the proposed NSSA and that it is not the backbone area.

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 *instance-tag***
3. **area *area-id* nssa [no-redistribution] [default-information-originate] [route-map *map-name*] [no-summary] [translate type7 {always | never}] [suppress-fa]**
4. (Optional) **address-family ipv6 unicast**
5. (Optional) **area *area-id* default-cost *cost***
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospfv3 <i>instance-tag</i> Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	area <i>area-id</i> nssa [no-redistribution] [default-information-originate] [route-map <i>map-name</i>] [no-summary] [translate type7 {always never}] [suppress-fa] Example: switch(config-router)# area 0.0.0.10 nssa	Creates this area as an NSSA.
Step 4	address-family ipv6 unicast Example: switch(config-router)# address-family ipv6 unicast switch(config-router-af)#	(Optional) Enters IPv6 unicast address family mode.

	Command	Purpose
Step 5	area <i>area-id</i> default-cost <i>cost</i> Example: switch(config-router-af)# area 0.0.0.10 default-cost 25	(Optional) Sets the cost metric for the default summary route sent into this NSSA. The range is from 0 to 16777215.
Step 6	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to create an NSSA that blocks all summary route updates:

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# area 0.0.0.10 nssa no-summary
switch(config-router)# copy running-config startup-config
```

This example shows how to create an NSSA that generates a default route:

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# area 0.0.0.10 nssa default-info-originate
switch(config-router)# copy running-config startup-config
```

This example shows how to create an NSSA that filters external routes and blocks all summary route updates:

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# area 0.0.0.10 nssa route-map ExternalFilter no-summary
switch(config-router)# copy running-config startup-config
```

This example shows how to create an NSSA that always translates Type-7 LSAs to AS External (type 5) LSAs:

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# area 0.0.0.10 nssa translate type 7 always
switch(config-router)# copy running-config startup-config
```

This example shows how to create an NSSA that blocks all summary route updates:

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# area 0.0.0.10 nssa no-summary
switch(config-router)# copy running-config startup-config
```

Configuring Multi-Area Adjacency

You can add more than one area to an existing OSPFv3 interface. The additional logical interfaces support multi-area adjacency.

BEFORE YOU BEGIN

You must enable OSPFv3 (see the [“Enabling OSPFv3”](#) section on page 5-14).

Ensure that you have configured a primary area for the interface (see the “Configuring Networks in OSPFv3” section on page 5-17)

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ipv6 router ospfv3** *instance-tag multi-area area-id*
4. (Optional) **show ipv6 ospfv3** *instance-tag interface interface-type slot/port*
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	ipv6 router ospfv3 <i>instance-tag multi-area area-id</i> Example: switch(config-if)# ipv6 router ospfv3 201 multi-area 3	Adds the interface to another area.
Step 4	show ipv6 ospfv3 <i>instance-tag interface interface-type slot/port</i> Example: switch(config-if)# show ipv6 ospfv3 201 interface ethernet 1/2	(Optional) Displays OSPFv3 information.
Step 5	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to add a second area to an OSPFv3 interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ipv6 address 2001:0DB8::1/48
switch(config-if)# ipv6 ospfv3 201 area 0.0.0.10
switch(config-if)# ipv6 ospfv3 201 multi-area 20
switch(config-if)# copy running-config startup-config
```

Configuring Virtual Links

A virtual link connects an isolated area to the backbone area through an intermediate area. See the [“Virtual Links” section on page 5-10](#). You can configure the following optional parameters for a virtual link:

- Authentication—Sets simple password or MD5 message digest authentication and associated keys.
- Dead interval—Sets the time that a neighbor waits for a Hello packet before declaring the local router as dead and tearing down adjacencies.
- Hello interval—Sets the time between successive Hello packets.
- Retransmit interval—Sets the estimated time between successive LSAs.
- Transmit delay—Sets the estimated time to transmit an LSA to a neighbor.



Note

You must configure the virtual link on both routers involved before the link becomes active.

BEFORE YOU BEGIN

You must enable OSPF (see the [“Enabling OSPFv3” section on page 5-14](#)).

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 *instance-tag***
3. **area *area-id* virtual-link *router-id***
4. (Optional) **show ipv6 ospfv3 virtual-link [brief]**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospfv3 <i>instance-tag</i> Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	area <i>area-id</i> virtual-link <i>router-id</i> Example: switch(config-router)# area 0.0.0.10 virtual-link 2001:0DB8::1 switch(config-router-vlink)#	Creates one end of a virtual link to a remote router. You must create the virtual link on that remote router to complete the link.

	Command	Purpose
Step 4	show ipv6 ospfv3 virtual-link [brief] Example: switch(config-if)# show ipv6 ospfv3 virtual-link	(Optional) Displays OSPFv3 virtual link information.
Step 5	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

You can configure the following optional commands in virtual link configuration mode:

Command	Purpose
dead-interval <i>seconds</i> Example: switch(config-router-vlink)# dead-interval 50	(Optional) Configures the OSPFv3 dead interval, in seconds. The range is from 1 to 65535. The default is four times the hello interval, in seconds.
hello-interval <i>seconds</i> Example: switch(config-router-vlink)# hello-interval 25	(Optional) Configures the OSPFv3 hello interval, in seconds. The range is from 1 to 65535. The default is 10 seconds.
retransmit-interval <i>seconds</i> Example: switch(config-router-vlink)# retransmit-interval 50	(Optional) Configures the OSPFv3 retransmit interval, in seconds. The range is from 1 to 65535. The default is 5.
transmit-delay <i>seconds</i> Example: switch(config-router-vlink)# transmit-delay 2	(Optional) Configures the OSPFv3 transmit-delay, in seconds. The range is from 1 to 450. The default is 1.

These examples show how to create a simple virtual link between two ABRs:

Configuration for ABR 1 (router ID 2001:0DB8::1) is as follows:

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# area 0.0.0.10 virtual-link 2001:0DB8::10
switch(config-router)# copy running-config startup-config
```

Configuration for ABR 2 (router ID 2001:0DB8::10) is as follows:

```
switch# configure terminal
switch(config)# router ospf 101
switch(config-router)# area 0.0.0.10 virtual-link 2001:0DB8::1
switch(config-router)# copy running-config startup-config
```


Configuring Redistribution

You can redistribute routes learned from other routing protocols into an OSPFv3 autonomous system through the ASBR.

You can configure the following optional parameters for route redistribution in OSPF:

- **Default information originate**—Generates an AS External (type 5) LSA for a default route to the external autonomous system.



Note Default information originate ignores **match** statements in the optional route map.

- **Default metric**—Sets all redistributed routes to the same cost metric.



Note If you redistribute static routes, Cisco NX-OS also redistributes the default static route.

BEFORE YOU BEGIN

Create the necessary route maps used for redistribution.

You must enable OSPF (see the [“Enabling OSPFv3” section on page 5-14](#)).

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 *instance-tag***
3. **address-family ipv6 unicast**
4. **redistribute {bgp *id* | direct | isis *id* | rip *id* | static} route-map *map-name***
5. **default-information originate [always] [route-map *map-name*]**
6. **default-metric *cost***
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospfv3 <i>instance-tag</i> Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.

	Command	Purpose
Step 3	address-family ipv6 unicast Example: <pre>switch(config-router)# address-family ipv6 unicast switch(config-router-af)#</pre>	Enters IPv6 unicast address family mode.
Step 4	redistribute {bgp id direct isis id rip id static} route-map map-name Example: <pre>switch(config-router-af)# redistribute bgp route-map FilterExternalBGP</pre>	Redistributes the selected protocol into OSPFv3 through the configured route map. Note If you redistribute static routes, Cisco NX-OS also redistributes the default static route.
Step 5	default-information originate [always] [route-map map-name] Example: <pre>switch(config-router-af)# default-information-originate route-map DefaultRouteFilter</pre>	Creates a default route into this OSPFv3 domain if the default route exists in the RIB. Use the following optional keywords: <ul style="list-style-type: none"> always —Always generates the default route of 0.0.0. even if the route does not exist in the RIB. route-map—Generates the default route if the route map returns true. Note This command ignores match statements in the route map.
Step 6	default-metric cost Example: <pre>switch(config-router-af)# default-metric 25</pre>	Sets the cost metric for the redistributed routes. The range is from 1 to 16777214. This command does not apply to directly connected routes. Use a route map to set the default metric for directly connected routes.
Step 7	copy running-config startup-config Example: <pre>switch(config-router)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

This example shows how to redistribute the Border Gateway Protocol (BGP) into OSPFv3:

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# address-family ipv6 unicast
switch(config-router-af)# redistribute bgp route-map FilterExternalBGP
switch(config-router-af)# copy running-config startup-config
```

Limiting the Number of Redistributed Routes

Route redistribution can add many routes to the OSPFv3 route table. You can configure a maximum limit to the number of routes accepted from external protocols. OSPFv3 provides the following options to configure redistributed route limits:

- Fixed limit—Logs a message when OSPFv3 reaches the configured maximum. OSPFv3 does not accept any more redistributed routes. You can optionally configure a threshold percentage of the maximum where OSPFv3 logs a warning when that threshold is passed.
- Warning only—Logs a warning only when OSPFv3 reaches the maximum. OSPFv3 continues to accept redistributed routes.

- **Withdraw**—Starts the configured timeout period when OSPFv3 reaches the maximum. After the timeout period, OSPFv3 requests all redistributed routes if the current number of redistributed routes is less than the maximum limit. If the current number of redistributed routes is at the maximum limit, OSPFv3 withdraws all redistributed routes. You must clear this condition before OSPFv3 accepts more redistributed routes. You can optionally configure the timeout period.

BEFORE YOU BEGIN

You must enable OSPF (see the [“Enabling OSPFv3” section on page 5-14](#)).

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 *instance-tag***
3. **address-family ipv6 unicast**
4. **redistribute {*bgp id* | *direct* | *isis id* | *rip id* | *static*} route-map *map-name***
5. **redistribute maximum-prefix *max* [*threshold*] [*warning-only* | *withdraw* [*num-retries* *timeout*]]**
6. (Optional) **show running-config ospfv3**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospfv3 <i>instance-tag</i> Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	address-family ipv6 unicast Example: switch(config-router)# address-family ipv6 unicast switch(config-router-af)#	Enters IPv6 unicast address family mode.
Step 4	redistribute {<i>bgp id</i> <i>direct</i> <i>isis id</i> <i>rip id</i> <i>static</i>} route-map <i>map-name</i> Example: switch(config-router-af)# redistribute bgp route-map FilterExternalBGP	Redistributes the selected protocol into OSPFv3 through the configured route map.

	Command	Purpose
Step 5	redistribute maximum-prefix <i>max</i> <i>[threshold] [warning-only withdraw</i> <i>[num-retries timeout]]</i> Example: switch(config-router)# redistribute maximum-prefix 1000 75 warning-only	Specifies a maximum number of prefixes that OSPFv2 distributes. The range is from 0 to 65536. Optionally, specifies the following: <ul style="list-style-type: none"> threshold—Percent of maximum prefixes that triggers a warning message. warning-only—Logs an warning message when the maximum number of prefixes is exceeded. withdraw—Withdraws all redistributed routes and optionally tries to retrieve the redistributed routes. The <i>num-retries</i> range is from 1 to 12. The <i>timeout</i> range is from 60 to 600 seconds. The default is 300 seconds.
Step 6	show running-config ospfv3 Example: switch(config-router)# show running-config ospf	(Optional) Displays the OSPFv3 configuration.
Step 7	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to limit the number of redistributed routes into OSPF:

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# address-family ipv6 unicast
switch(config-router-af)# redistribute bgp route-map FilterExternalBGP
switch(config-router-af)# redistribute maximum-prefix 1000 75
```

Configuring Route Summarization

You can configure route summarization for inter-area routes by configuring an address range that is summarized. You can also configure route summarization for external, redistributed routes by configuring a summary address for those routes on an ASBR. For more information, see the [“Route Summarization”](#) section on page 5-10.

BEFORE YOU BEGIN

You must enable OSPF (see the [“Enabling OSPFv3”](#) section on page 5-14).

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 instance-tag**
3. **address-family ipv6 unicast**
4. **area area-id range ipv6-prefix/length [no-advertise] [cost cost]**

or

5. **summary-address** *ipv6-prefix/length* [**no-advertise**] [**tag** *tag*]
6. (Optional) **show ipv6 ospfv3 summary-address**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospfv3 <i>instance-tag</i> Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	address-family ipv6 unicast Example: switch(config-router)# address-family ipv6 unicast switch(config-router-af)#	Enters IPv6 unicast address family mode.
Step 4	area <i>area-id</i> range <i>ipv6-prefix/length</i> [no-advertise] [cost <i>cost</i>] Example: switch(config-router-af)# area 0.0.0.10 range 2001:0DB8::/48 advertise	Creates a summary address on an ABR for a range of addresses <i>and</i> optionally advertises this summary address in a Inter-Area Prefix (type 3) LSA. The <i>cost</i> range is from 0 to 16777215.
Step 5	summary-address <i>ipv6-prefix/length</i> [no-advertise] [tag <i>tag</i>] Example: switch(config-router-af)# summary-address 2001:0DB8::/48 tag 2	Creates a summary address on an ASBR for a range of addresses and optionally assigns a tag for this summary address that can be used for redistribution with route maps.
Step 6	show ipv6 ospfv3 summary-address Example: switch(config-router)# show ipv6 ospfv3 summary-address	(Optional) Displays information about OSPFv3 summary addresses.
Step 7	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to create summary addresses between areas on an ABR:

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# address-family ipv6 unicast
switch(config-router)# area 0.0.0.10 range 2001:0DB8::/48
switch(config-router)# copy running-config startup-config
```

This example shows how to create summary addresses on an ASBR:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# address-family ipv6 unicast
switch(config-router)# summary-address 2001:0DB8::/48
switch(config-router)# copy running-config startup-config
```

Modifying the Default Timers

OSPFv3 includes a number of timers that control the behavior of protocol messages and shortest path first (SPF) calculations. OSPFv3 includes the following optional timer parameters:

- LSA arrival time—Sets the minimum interval allowed between LSAs arriving from a neighbor. LSAs that arrive faster than this time are dropped.
- Pacing LSAs—Sets the interval at which LSAs are collected into a group and refreshed, checksummed, or aged. This timer controls how frequently LSA updates occur and optimizes how many are sent in an LSA update message (see the [“Flooding and LSA Group Pacing” section on page 5-7](#)).
- Throttle LSAs—Sets rate limits for generating LSAs. This timer controls how frequently LSAs are generated after a topology change occurs.
- Throttle SPF calculation—Controls how frequently the SPF calculation is run.

At the interface level, you can also control the following timers:

- Retransmit interval—Sets the estimated time between successive LSAs.
- Transmit delay—Sets the estimated time to transmit an LSA to a neighbor.

See the [“Configuring Networks in OSPFv3” section on page 5-17](#) for information on the hello interval and dead timer.

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3** *instance-tag*
3. **timers lsa-arrival** *msec*
4. **timers lsa-group-pacing** *seconds*
5. **timers throttle lsa** *start-time hold-interval max-time*
6. **address-family ipv6 unicast**
7. **timers throttle spf** *delay-time hold-time max-time*
8. **interface** *type slot/port*
9. **ospfv3 retransmit-interval** *seconds*
10. **ospfv3 transmit-delay** *seconds*
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospfv3 instance-tag Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	timers lsa-arrival msec Example: switch(config-router)# timers lsa-arrival 2000	Sets the LSA arrival time in milliseconds. The range is from 10 to 600000. The default is 1000 milliseconds.
Step 4	timers lsa-group-pacing seconds Example: switch(config-router)# timers lsa-group-pacing 200	Sets the interval in seconds for grouping LSAs. The range is from 1 to 1800. The default is 10 seconds.
Step 5	timers throttle lsa start-time hold-interval max-time Example: switch(config-router)# timers throttle lsa network 350 5000 6000	Sets the rate limit in milliseconds for generating LSAs. You can configure the following timers: <i>start-time</i> —The range is from 50 to 5000 milliseconds. The default value is 50 milliseconds. <i>hold-interval</i> —The range is from 50 to 30,000 milliseconds. The default value is 5000 milliseconds. <i>max-time</i> —The range is from 50 to 30,000 milliseconds. The default value is 5000 milliseconds.
Step 6	address-family ipv6 unicast Example: switch(config-router)# address-family ipv6 unicast switch(config-router-af)#	Enters IPv6 unicast address family mode.
Step 7	timers throttle spf delay-time hold-time max-time Example: switch(config-router)# timers throttle spf 3000 2000 5000	Sets the SPF best-path schedule initial delay time, minimum hold time, and maximum wait time in milliseconds between SPF best-path calculations. The range is from 1 to 600000 milliseconds. The default values are a 200-ms delay time, 1000-ms hold time, and 5000-ms wait time.
Step 8	interface type slot/port Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 9	ospfv3 retransmit-interval seconds Example: switch(config-if)# ospfv3 retransmit-interval 30	Sets the estimated time in seconds between LSAs transmitted from this interface. The range is from 1 to 65535. The default is 5.

	Command	Purpose
Step 10	ospfv3 transmit-delay <i>seconds</i> Example: switch(config-if)# ospfv3 transmit-delay 600 switch(config-if)#	Sets the estimated time in seconds to transmit an LSA to a neighbor. The range is from 1 to 450. The default is 1.
Step 11	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to control LSA flooding with the lsa-group-pacing option:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# timers lsa-group-pacing 300
switch(config-router)# copy running-config startup-config
```

Configuring Graceful Restart

Graceful restart is enabled by default. You can configure the following optional parameters for graceful restart in an OSPFv3 instance:

- Grace period—Configures how long neighbors should wait after a graceful restart has started before tearing down adjacencies.
- Helper mode disabled—Disables helper mode on the local OSPFv3 instance. OSPFv3 does not participate in the graceful restart of a neighbor.
- Planned graceful restart only—Configures OSPFv3 to support graceful restart only in the event of a planned restart.

BEFORE YOU BEGIN

You must enable OSPFv3 (see the [“Enabling OSPFv3”](#) section on page 5-14).

Ensure that all neighbors are configured for graceful restart with matching optional parameters set.

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3** *instance-tag*
3. **graceful-restart**
4. **graceful-restart grace-period** *seconds*
5. **graceful-restart helper-disable**
6. **graceful-restart planned-only**
7. (Optional) **show ipv6 ospfv3** *instance-tag*
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospfv3 instance-tag Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	graceful-restart Example: switch(config-router)# graceful-restart	Enables graceful restart. A graceful restart is enabled by default.
Step 4	graceful-restart grace-period seconds Example: switch(config-router)# graceful-restart grace-period 120	Sets the grace period, in seconds. The range is from 5 to 1800. The default is 60 seconds.
Step 5	graceful-restart helper-disable Example: switch(config-router)# graceful-restart helper-disable	Disables helper mode. Enabled by default.
Step 6	graceful-restart planned-only Example: switch(config-router)# graceful-restart planned-only	Configures graceful restart for planned restarts only.
Step 7	show ipv6 ospfv3 instance-tag Example: switch(config-if)# show ipv6 ospfv3 201	(Optional) Displays OSPFv3 information.
Step 8	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to enable graceful restart if it has been disabled and set the grace period to 120 seconds:

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# graceful-restart
switch(config-router)# graceful-restart grace-period 120
switch(config-router)# copy running-config startup-config
```

Restarting an OSPFv3 Instance

You can restart an OSPFv3 instance. This action clears all neighbors for the instance.

To restart an OSPFv3 instance and remove all associated neighbors, use the following command:

Command	Purpose
restart ospfv3 instance-tag	Restarts the OSPFv3 instance and removes all neighbors.
Example: switch(config)# restart ospfv3 201	

Configuring OSPFv3 with Virtualization

You can configure multiple OSPFv3 instances in each VDC. You can also create multiple VRFs within each VDC and use the same or multiple OSPFv3 instances in each VRF. You assign an OSPFv3 interface to a VRF.

**Note**

Configure all other parameters for an interface after you configure the VRF for an interface. Configuring a VRF for an interface deletes all the configuration for that interface.

BEFORE YOU BEGIN

Create the VDCs.

You must enable OSPF (see the [“Enabling OSPFv3”](#) section on page 5-14).

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf_name*
3. **router ospfv3** *instance-tag*
4. **vrf** *vrf-name*
5. (Optional) **maximum-paths** *paths*
6. **interface** *type slot/port*
7. **vrf member** *vrf-name*
8. **ipv6 address** *ipv6-prefix/length*
9. **ipv6 ospfv3** *instance-tag* **area** *area-id*
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	vrf context <i>vrf-name</i> Example: switch(config)# vrf context RemoteOfficeVRF switch(config-vrf)#	Creates a new VRF and enters VRF configuration mode.
Step 3	router ospfv3 <i>instance-tag</i> Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 4	vrf <i>vrf-name</i> Example: switch(config-router)# vrf RemoteOfficeVRF switch(config-router-vrf)#	Enters VRF configuration mode.
Step 5	maximum-paths <i>paths</i> Example: switch(config-router-vrf)# maximum-paths 4	(Optional) Configures the maximum number of equal OSPFv3 paths to a destination in the route table for this VRF. Use this command for load balancing.
Step 6	interface <i>type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 7	vrf member <i>vrf-name</i> Example: switch(config-if)# vrf member RemoteOfficeVRF	Adds this interface to a VRF.
Step 8	ipv6 address <i>ipv6-prefix/length</i> Example: switch(config-if)# ipv6 address 2001:0DB8::1/48	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 9	ipv6 ospfv3 <i>instance-tag area area-id</i> Example: switch(config-if)# ipv6 ospfv3 201 area 0	Assigns this interface to the OSPFv3 instance and area configured.
Step 10	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to create a VRF and add an interface to the VRF:

```
switch# configure terminal
switch(config)# vrf context NewVRF
switch(config-vrf)# exit
switch(config)# router ospfv3 201
switch(config-router)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# vrf member NewVRF
switch(config-if)# ipv6 address 2001:0DB8::1/48
switch(config-if)# ipv6 ospfv3 201 area 0
switch(config-if)# copy running-config startup-config
```

Verifying the OSPFv3 Configuration

To display the OSPFv3 configuration, perform one of the following tasks:

Command	Purpose
show ipv6 ospfv3	Displays the OSPFv3 configuration.
show ipv6 ospfv3 border-routers	Displays the internal OSPF routing table entries to an ABR and ASBR.
show ipv6 ospfv3 database	Displays lists of information related to the OSPFv3 database for a specific router.
show ipv6 ospfv3 interface <i>type number</i> [vrf { <i>vrf-name</i> all default management }]	Displays the OSPFv3 interface configuration.
show ipv6 ospfv3 neighbors	Displays the neighbor information. Use the clear ospfv3 neighbors command to remove adjacency with all neighbors.
show ipv6 ospfv3 request-list	Displays a list of LSAs requested by a router.
show ipv6 ospfv3 retransmission-list	Displays a list of LSAs waiting to be retransmitted.
show ipv6 ospfv3 summary-address	Displays a list of all summary address redistribution information configured under an OSPFv3 instance.
show running-configuration ospfv3	Displays the current running OSPFv3 configuration.

Monitoring OSPFv3

To display OSPFv3 statistics, use the following commands:

Command	Purpose
show ipv6 ospfv3 memory	Displays the OSPFv3 memory usage statistics.
show ipv6 ospfv3 policy statistics area <i>area-id</i> filter-list { <i>in</i> <i>out</i> } [vrf { <i>vrf-name</i> <i>all</i> <i>default</i> <i>management</i> }]	Displays the OSPFv3 route policy statistics for an area.
show ipv6 ospfv3 policy statistics redistribute { <i>bgp id</i> <i>direct</i> <i>isis id</i> <i>rip id</i> <i>static</i> } [vrf { <i>vrf-name</i> <i>all</i> <i>default</i> <i>management</i> }]	Displays the OSPFv3 route policy statistics.
show ipv6 ospfv3 statistics [vrf { <i>vrf-name</i> <i>all</i> <i>default</i> <i>management</i> }]	Displays the OSPFv3 event counters.
show ipv6 ospfv3 traffic [<i>interface-type</i> <i>number</i>] [vrf { <i>vrf-name</i> <i>all</i> <i>default</i> <i>management</i> }]	Displays the OSPFv3 packet counters.

Configuration Examples for OSPFv3

This example shows how to configure OSPFv3:

```
feature ospfv3
router ospfv3 201
  router-id 290.0.2.1

interface ethernet 1/2
  ipv6 address 2001:0DB8::1/48
  ipv6 ospfv3 201 area 0.0.0.10
```

Related Topics

The following topics can give more information on OSPF:

- [Chapter 4, “Configuring OSPFv2”](#)
- [Chapter 16, “Configuring Route Policy Manager”](#)

Additional References

For additional information related to implementing OSPF, see the following sections:

- [Related Documents, page 5-42](#)
- [MIBs, page 5-42](#)

Related Documents

Related Topic	Document Title
OSPFv3 CLI commands	<i>Cisco Nexus 3000 Series NX-OS Unicast Routing Command Reference, Release 6.x</i>

MIBs

MIBs	MIBs Link
<ul style="list-style-type: none"> OSPF-MIB OSPF-TRAP-MIB 	<p>To locate and download MIBs, go to the following URL:</p> <p>http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</p>



Configuring EIGRP

This chapter describes how to configure the Enhanced Interior Gateway Routing Protocol (*EIGRP*) on the Cisco NX-OS switch.

This chapter includes the following sections:

- [Information About EIGRP, page 6-1](#)
- [Licensing Requirements for EIGRP, page 6-7](#)
- [Prerequisites for EIGRP, page 6-7](#)
- [Guidelines and Limitations, page 6-7](#)
- [Default Settings, page 6-8](#)
- [Configuring Basic EIGRP, page 6-8](#)
- [Configuring Advanced EIGRP, page 6-13](#)
- [Configuring Virtualization for EIGRP, page 6-25](#)
- [Verifying the EIGRP Configuration, page 6-27](#)
- [Displaying EIGRP Statistics, page 6-27](#)
- [Configuration Examples for EIGRP, page 6-28](#)
- [Related Topics, page 6-28](#)
- [Additional References, page 6-28](#)

Information About EIGRP

EIGRP combines the benefits of distance vector protocols with the features of link-state protocols. EIGRP sends out periodic hello messages for neighbor discovery. Once EIGRP learns a new neighbor, it sends a one-time update of all the local EIGRP routes and route metrics. The receiving EIGRP router calculates the route distance based on the received metrics and the locally assigned cost of the link to that neighbor. After this initial full route table update, EIGRP sends incremental updates to only those neighbors affected by the route change. This process speeds convergence and minimizes the bandwidth used by EIGRP.

This section includes the following topics:

- [EIGRP Components, page 6-2](#)
- [EIGRP Route Updates, page 6-3](#)
- [Advanced EIGRP, page 6-4](#)

EIGRP Components

EIGRP has the following basic components:

- [Reliable Transport Protocol, page 6-2](#)
- [Neighbor Discovery and Recovery, page 6-2](#)
- [Diffusing Update Algorithm, page 6-2](#)

Reliable Transport Protocol

The *Reliable Transport Protocol* guarantees ordered delivery of EIGRP packets to all neighbors. (See the “[Neighbor Discovery and Recovery](#)” section on [page 6-2](#).) The Reliable Transport Protocol supports the transmission of unicast packets. This provision helps to ensure that the convergence time remains low for various speed links. See the “[Configuring Advanced EIGRP](#)” section on [page 6-13](#) for details about modifying the default timers that control the unicast packet transmissions.

The Reliable Transport Protocol includes the following message types:

- Hello—Used for neighbor discovery and recovery. By default, EIGRP sends a periodic multicast hello message on the local network at the configured *hello interval*. By default, the hello interval is 5 seconds.
- Acknowledgement—Verifies reliable reception of Updates, Queries, and Replies.
- Updates—Sends to affected neighbors when routing information changes. Updates include the route destination, address mask, and route metrics such as delay and bandwidth. The update information is stored in the EIGRP topology table.
- Queries and Replies—Sent as necessary as part of the Diffusing Update Algorithm used by EIGRP.

Neighbor Discovery and Recovery

EIGRP uses the hello messages from the Reliable Transport Protocol to discover neighboring EIGRP routers on directly attached networks. EIGRP adds neighbors to the neighbor table. The information in the neighbor table includes the neighbor address, the interface it was learned on, and the *hold time*, which indicates how long EIGRP should wait before declaring a neighbor unreachable. By default, the hold time is three times the hello interval or 15 seconds.

EIGRP sends a series of Update messages to new neighbors to share the local EIGRP routing information. This route information is stored in the EIGRP topology table. After this initial transmission of the full EIGRP route information, EIGRP sends Update messages only when a routing change occurs. These Update messages contain only the new or changed information and are sent only to the neighbors affected by the change. See the “[EIGRP Route Updates](#)” section on [page 6-3](#).

EIGRP also uses the hello messages as a keepalive to its neighbors. As long as hello messages are received, Cisco NX-OS can determine that a neighbor is alive and functioning.

Diffusing Update Algorithm

The *Diffusing Update Algorithm* (DUAL) calculates the routing information based on the destination networks in the topology table. The topology table includes the following information:

- IPv4 address/mask—The network address and network mask for this destination.
- Successors—The IP address and local interface connection for all *feasible successors* or neighbors that advertise a shorter distance to the destination than the current *feasible distance*.

- Feasibility distance (FD)—The lowest calculated distance to the destination. The feasibility distance is the sum of the advertised distance from a neighbor plus the cost of the link to that neighbor.

DUAL uses the distance metric to select efficient, loop-free paths. DUAL selects routes to insert into the unicast Routing Information Base (RIB) based on feasible successors. When a topology change occurs, DUAL looks for feasible successors in the topology table. If there are feasible successors, DUAL selects the feasible successor with the lowest feasible distance and inserts that into the unicast RIB, avoiding unnecessary recomputation.

When there are no feasible successors but there are neighbors advertising the destination, DUAL transitions from the passive state to the active state and triggers a recomputation to determine a new successor or next-hop router to the destination. The amount of time required to recompute the route affects the convergence time. EIGRP sends Query messages to all neighbors, searching for feasible successors. Neighbors that have a feasible successor send a Reply message with that information. Neighbors that do not have feasible successors trigger a DUAL recomputation.

EIGRP Route Updates

When a topology change occurs, EIGRP sends an Update message with only the changed routing information to affected neighbors. This Update message includes the distance information to the new or updated network destination.

The distance information in EIGRP is represented as a composite of available route metrics, including bandwidth, delay, load utilization, and link reliability. Each metric has an associated weight that determines if the metric is included in the distance calculation. You can configure these metric weights. You can fine-tune link characteristics to achieve optimal paths, but we recommend that you use the default settings for most configurable metrics.

This section includes the following topics:

- [Internal Route Metrics, page 6-3](#)
- [External Route Metrics, page 6-4](#)
- [EIGRP and the Unicast RIB, page 6-4](#)

Internal Route Metrics

Internal routes are routes that occur between neighbors within the same EIGRP autonomous system. These routes have the following metrics:

- Next hop—The IP address of the next-hop router.
- Delay—The sum of the delays configured on the interfaces that make up the route to the destination network. Configured in tens of microseconds.
- Bandwidth—The calculation from the lowest configured bandwidth on an interface that is part of the route to the destination.



Note We recommend that you use the default bandwidth value. This bandwidth parameter is also used by EIGRP.

- MTU—The smallest maximum transmission unit value along the route to the destination.
- Hop count—The number of hops or routers that the route passes through to the destination. This metric is not directly used in the DUAL computation.

- Reliability—An indication of the reliability of the links to the destination.
- Load—An indication of how much traffic is on the links to the destination.

By default, EIGRP uses the bandwidth and delay metrics to calculate the distance to the destination. You can modify the metric weights to include the other metrics in the calculation.

External Route Metrics

External routes are routes that occur between neighbors in different EIGRP autonomous systems. These routes have the following metrics:

- Next hop—The IP address of the next-hop router.
- Router ID—The router ID of the router that redistributed this route into EIGRP.
- AS Number—The autonomous system number of the destination.
- Protocol ID—A code that represents the routing protocol that learned the destination route.
- Tag—An arbitrary tag that can be used for route maps.
- Metric—The route metric for this route from the external routing protocol.

EIGRP and the Unicast RIB

EIGRP adds all learned routes to the EIGRP topology table and the unicast RIB. When a topology change occurs, EIGRP uses these routes to search for a feasible successor. EIGRP also listens for notifications from the unicast RIB for changes in any routes redistributed to EIGRP from another routing protocol.

Advanced EIGRP

You can use the advanced features of EIGRP to optimize your EIGRP configuration.

This section includes the following topics:

- [Address Families, page 6-4](#)
- [Authentication, page 6-5](#)
- [Stub Routers, page 6-5](#)
- [Route Summarization, page 6-6](#)
- [Route Redistribution, page 6-6](#)
- [Load Balancing, page 6-6](#)
- [Split Horizon, page 6-6](#)
- [Virtualization Support, page 6-7](#)

Address Families

EIGRP supports the IPv4 address family.

Address family configuration mode includes the following EIGRP features:

- Authentication
- AS number

- Default route
- Metrics
- Distance
- Graceful restart
- Logging
- Load balancing
- Redistribution
- Router ID
- Stub router
- Timers

You cannot configure the same feature in more than one configuration mode. For example, if you configure the default metric in router configuration mode, you cannot configure the default metric in address family mode.

Authentication

You can configure authentication on EIGRP messages to prevent unauthorized or invalid routing updates in your network. EIGRP authentication supports MD5 authentication digest.

You can configure the EIGRP authentication per virtual routing and forwarding (VRF) instance or interface using keychain management for the authentication keys. Keychain management allows you to control changes to the authentication keys used by MD5 authentication digest.

For MD5 authentication, you configure a password that is shared at the local router and all remote EIGRP neighbors. When an EIGRP message is created, Cisco NX-OS creates an MD5 one-way message digest based on the message itself and the encrypted password and sends this digest along with the EIGRP message. The receiving EIGRP neighbor validates the digest using the same encrypted password. If the message has not changed, the calculation is identical and the EIGRP message is considered valid.

MD5 authentication also includes a sequence number with each EIGRP message that is used to ensure that no message is replayed in the network.

Stub Routers

You can use the EIGRP stub routing feature to improve network stability, reduce resource usage, and simplify stub router configuration. Stub routers connect to the EIGRP network through a remote router. See the [“Stub Routing” section on page 1-7](#).

When using EIGRP stub routing, you need to configure the distribution and remote routers to use EIGRP and configure only the remote router as a stub. EIGRP stub routing does not automatically enable summarization on the distribution router. In most cases, you need to configure summarization on the distribution routers.

Without EIGRP stub routing, even after the routes that are sent from the distribution router to the remote router have been filtered or summarized, a problem might occur. For example, if a route is lost somewhere in the corporate network, EIGRP could send a query to the distribution router. The distribution router could then send a query to the remote router even if routes are summarized. If a problem communicating over the WAN link between the distribution router and the remote router occurs, EIGRP could get stuck in active condition and cause instability elsewhere in the network. EIGRP stub routing allows you to prevent queries to the remote router.

Route Summarization

You can configure a summary aggregate address for a specified interface. Route summarization simplifies route tables by replacing a number of more-specific addresses with an address that represents all the specific addresses. For example, you can replace 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24 with one summary address, 10.1.0.0/16.

If more specific routes are in the routing table, EIGRP advertises the summary address from the interface with a metric equal to the minimum metric of the more specific routes.

**Note**

EIGRP does not support automatic route summarization.

Route Redistribution

You can use EIGRP to redistribute direct routes, static routes, routes learned by other EIGRP autonomous systems, or routes from other protocols. You configure route map with the redistribution to control which routes are passed into EIGRP. A route map allows you to filter routes based on attributes such as the destination, origination protocol, route type, route tag, and so on. See [Chapter 16, “Configuring Route Policy Manager.”](#)

You also configure the default metric that is used for all imported routes into EIGRP.

Load Balancing

You can use load balancing to allow a router to distribute traffic over all the router network ports that are the same distance from the destination address. Load balancing increases the utilization of network segments, which increases effective network bandwidth.

Cisco NX-OS supports the Equal Cost Multiple Paths (ECMP) feature with up to 16 equal-cost paths in the EIGRP route table and the unicast RIB. You can configure EIGRP to load balance traffic across some or all of those paths.

**Note**

EIGRP in Cisco NX-OS does not support unequal cost load balancing.

Split Horizon

You can use split horizon to ensure that EIGRP never advertises a route out of the interface where it was learned.

Split horizon is a method that controls the sending of EIGRP update and query packets. When you enable split horizon on an interface, Cisco NX-OS does not send update and query packets for destinations that were learned from this interface. Controlling update and query packets in this manner reduces the possibility of routing loops.

Split horizon with poison reverse configures EIGRP to advertise a learned route as unreachable back through that the interface that EIGRP learned the route from.

EIGRP uses split horizon or split horizon with poison reverse in the following scenarios:

- Exchanging topology tables for the first time between two routers in startup mode.
- Advertising a topology table change.
- Sending a query message.

By default, the split horizon feature is enabled on all interfaces.

Virtualization Support

Cisco NX-OS supports multiple instances of the EIGRP protocol that runs on the same system. EIGRP supports virtual routing and forwarding (VRF) instance. By default, Cisco NX-OS places you in the default VRF unless you specifically configure another VRF. See [Chapter 14, “Configuring Layer 3 Virtualization.”](#)

By default, every instance uses the same system router ID. You can optionally configure a unique router ID for each instance.

Licensing Requirements for EIGRP

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	Full EIGRP requires a Enterprise Services license, however EIGRP stub requires a LAN Base Services license. For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the <i>Cisco NX-OS Licensing Guide</i> .

Prerequisites for EIGRP

EIGRP has the following prerequisites:

- You must enable the EIGRP feature (see the [“Enabling the EIGRP Feature”](#) section on page 6-9).

Guidelines and Limitations

EIGRP has the following configuration guidelines and limitations:

- A metric configuration (either through the default-metric configuration option or through a route map) is required for redistribution from any other protocol, connected routes, or static routes (see [Chapter 16, “Configuring Route Policy Manager”](#)).
- For graceful restart, an NSF-aware router must be up and completely converged with the network before it can assist an NSF-capable router in a graceful restart operation.
- For graceful restart, neighboring switches participating in the graceful restart must be NSF-aware or NSF-capable.
- Cisco NX-OS EIGRP is compatible with EIGRP in the Cisco IOS software.
- Do not change the metric weights without a good reason. If you change the metric weights, you must apply the change to all EIGRP routers in the same autonomous system.
- Consider using stubs for larger networks.
- Avoid redistribution between different EIGRP autonomous systems because the EIGRP vector metric will not be preserved.
- The **no ip next-hop-self** command does not guarantee reachability of the next-hop.

- The **ip passive-interface eigrp** command suppresses neighbors from forming.
- Cisco NX-OS does not support IGRP or connecting IGRP and EIGRP clouds.
- Autosummarization is not enabled by default.
- Cisco NX-OS supports only IP.

**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.

Default Settings

Table 6-1 lists the default settings for EIGRP parameters.

Table 6-1 *Default EIGRP Parameters*

Parameters	Default
Administrative distance	<ul style="list-style-type: none"> • Internal routes—90 • External routes—170
Bandwidth percent	50 percent
Default metric for redistributed routes	<ul style="list-style-type: none"> • bandwidth—100000 Kb/s • delay—100 (10 microsecond units) • reliability—255 • loading—1 • MTU—1500
EIGRP feature	Disabled
Hello interval	5 seconds
Hold time	15 seconds
Equal-cost paths	8
Metric weights	1 0 1 0 0
Next-hop address advertised	IP address of local interface
NSF convergence time	120
NSF route-hold time	240
NSF signal time	20
Redistribution	Disabled
Split horizon	Enabled

Configuring Basic EIGRP

This section includes the following topics:

- [Enabling the EIGRP Feature, page 6-9](#)

- [Creating an EIGRP Instance, page 6-10](#)
- [Restarting an EIGRP Instance, page 6-12](#)
- [Shutting Down an EIGRP Instance, page 6-12](#)
- [Shutting Down EIGRP on an Interface, page 6-13](#)

Enabling the EIGRP Feature

You must enable the EIGRP feature before you can configure EIGRP.

SUMMARY STEPS

1. **configure terminal**
2. **feature eigrp**
3. (Optional) **show feature**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	feature eigrp Example: switch(config)# feature eigrp	Enables the EIGRP feature.
Step 3	show feature Example: switch(config)# show feature	(Optional) Displays information about enabled features.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

Use the **no feature eigrp** command to disable the EIGRP feature and remove all associated configuration.

Command	Purpose
no feature eigrp Example: switch(config)# no feature eigrp	Disables the EIGRP feature and removes all associated configuration.

Creating an EIGRP Instance

You can create an EIGRP instance and associate an interface with that instance. You assign a unique autonomous system number for this EIGRP process (see the [“Autonomous Systems” section on page 1-5](#)). Routes are not advertised or accepted from other autonomous systems unless you enable route redistribution.

BEFORE YOU BEGIN

Ensure that you have enabled the EIGRP feature (see the [“Enabling the EIGRP Feature” section on page 6-9](#)).

EIGRP must be able to obtain a router ID (for example, a configured loopback address) or you must configure the router ID option.

If you configure an instance tag that does not qualify as an AS number, you must configure the AS number explicitly or this EIGRP instance will remain in the shutdown state.

SUMMARY STEPS

1. **configure terminal**
2. **router eigrp** *instance-tag*
3. (Optional) **autonomous-system** *as-number*
4. (Optional) **log-adjacency-changes**
5. (Optional) **log-neighbor-warnings** [*seconds*]
6. **interface** *interface-type slot/port*
7. **no switchport**
8. **ip router eigrp** *instance-tag*
9. **show ip eigrp interfaces**
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router eigrp <i>instance-tag</i> Example: switch(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 <i>instance-tag</i> that does not qualify as an AS number, you must use the autonomous-system command to configure the AS number explicitly or this EIGRP instance will remain in the shutdown state.

	Command	Purpose
Step 3	autonomous-system <i>as-number</i> Example: switch(config-router)# autonomous-system 33	(Optional) Configures a unique AS number for this EIGRP instance. The range is from 1 to 65535.
Step 4	log-adjacency-changes Example: switch(config-router)# log-adjacency-changes	(Optional). Generates a system message whenever an adjacency changes state. This command is enabled by default.
Step 5	log-neighbor-warnings [<i>seconds</i>] Example: switch(config-router)# log-neighbor-warnings	(Optional) Generates a system message whenever a neighbor warning occurs. You can configure the time between warning messages, from 1 to 65535, in seconds. The default is 10 seconds. This command is enabled by default.
Step 6	interface <i>interface-type slot/port</i> Example: switch(config-router)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode. Use ? to determine the slot and port ranges.
Step 7	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 8	ip router eigrp <i>instance-tag</i> Example: switch(config-if)# ip router eigrp Test1	Associates this interface with the configured EIGRP process. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
Step 9	show ip eigrp interfaces Example: switch(config-if)# show ip eigrp interfaces	Displays information about EIGRP interfaces.
Step 10	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

Use the **no router eigrp** command to remove the EIGRP process and the associated configuration.

Command	Purpose
no router eigrp <i>instance-tag</i> Example: switch(config)# no router eigrp Test1	Deletes the EIGRP process and all associated configuration.

**Note**

You should also remove any EIGRP commands configured in interface mode if you remove the EIGRP process.

This example shows how to create an EIGRP process and configure an interface for EIGRP:

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# ip router eigrp Test1
switch(config-if)# no shutdown
switch(config-if)# copy running-config startup-config
```

For more information about other EIGRP parameters, see the [“Configuring Advanced EIGRP”](#) section on page 6-13.

Restarting an EIGRP Instance

You can restart an EIGRP instance. This clears all neighbors for the instance.

To restart an EIGRP instance and remove all associated neighbors, use the following commands:

Command	Purpose
flush-routes Example: switch(config)# flush-routes	(Optional) Flushes all EIGRP routes in the unicast RIB when this EIGRP instance restarts.
restart eigrp instance-tag Example: switch(config)# restart eigrp Test1	Restarts the EIGRP instance and removes all neighbors. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.

Shutting Down an EIGRP Instance

You can gracefully shut down an EIGRP instance. This action moves all routes and adjacencies but preserves the EIGRP configuration.

To disable an EIGRP instance, use the following command in router configuration mode:

Command	Purpose
shutdown Example: switch(config-router)# shutdown	Disables this instance of EIGRP. The EIGRP router configuration remains.

Configuring a Passive Interface for EIGRP

You can configure a passive interface for EIGRP. A passive interface does not participate in EIGRP adjacency but the network address for the interface remains in the EIGRP topology table.

To configure a passive interface for EIGRP, use the following command in interface configuration mode:

Command	Purpose
<code>ip passive-interface eigrp instance-tag</code>	Suppresses EIGRP hellos, which prevents neighbors from forming and sending routing updates on an EIGRP interface. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.

Shutting Down EIGRP on an Interface

You can gracefully shut down EIGRP on an interface. This action removes all adjacencies and stops EIGRP traffic on this interface but preserves the EIGRP configuration.

To disable EIGRP on an interface, use the following command in interface configuration mode:

Command	Purpose
<pre>switch(config-if)# ip eigrp instance-tag shutdown</pre> <p>Example:</p> <pre>switch(config-router)# ip eigrp Test1 shutdown</pre>	Disables EIGRP on this interface. The EIGRP interface configuration remains. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.

Configuring Advanced EIGRP

This section includes the following topics:

- [Configuring Authentication in EIGRP, page 6-13](#)
- [Configuring EIGRP Stub Routing, page 6-16](#)
- [Configuring a Summary Address for EIGRP, page 6-16](#)
- [Redistributing Routes into EIGRP, page 6-17](#)
- [Limiting the Number of Redistributed Routes, page 6-19](#)
- [Configuring Load Balancing in EIGRP, page 6-21](#)
- [Adjusting the Interval Between Hello Packets and the Hold Time, page 6-22](#)
- [Disabling Split Horizon, page 6-23](#)
- [Tuning EIGRP, page 6-23](#)

Configuring Authentication in EIGRP

You can configure authentication between neighbors for EIGRP. See the [“Authentication” section on page 6-5](#).

You can configure EIGRP authentication for the EIGRP process or for individual interfaces. Interface EIGRP authentication configuration overrides the EIGRP process-level authentication configuration.

BEFORE YOU BEGIN

Ensure that you have enabled the EIGRP feature (see the [“Enabling the EIGRP Feature”](#) section on page 6-9).

Ensure that all neighbors for an EIGRP process share the same authentication configuration, including the shared authentication key.

Create a keychain for this authentication configuration.

SUMMARY STEPS

1. **configure terminal**
2. **router eigrp** *instance-tag*
3. **address-family ipv4 unicast**
4. **authentication keychain** *keychain*
5. **authentication mode md5**
6. **interface** *interface-type slot/port*
7. **no switchport**
8. **ip router eigrp** *instance-tag*
9. **ip authentication keychain eigrp** *instance-tag keychain*
10. **ip authentication mode eigrp** *instance-tag md5*
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router eigrp <i>instance-tag</i> Example: switch(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 <i>instance-tag</i> that does not qualify as an AS number, you must use the autonomous-system command to configure the AS number explicitly or this EIGRP instance will remain in the shutdown state.
Step 3	address-family ipv4 unicast Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	Enters the address-family configuration mode. This command is optional for IPv4.

	Command	Purpose
Step 4	authentication keychain <i>keychain</i> Example: switch(config-router-af)# authentication keychain routeKeys	Associates a keychain with this EIGRP process for this VRF. The keychain can be any case-sensitive, alphanumeric string up to 20 characters.
Step 5	authentication mode md5 Example: switch(config-router-af)# authentication mode md5	Configures MD5 message digest authentication mode for this VRF.
Step 6	interface <i>interface-type slot/port</i> Example: switch(config-router-af) interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode. Use ? to find the supported interfaces.
Step 7	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 8	ip router eigrp <i>instance-tag</i> Example: switch(config-if)# ip router eigrp Test1	Associates this interface with the configured EIGRP process. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
Step 9	ip authentication keychain eigrp <i>instance-tag keychain</i> Example: switch(config-if)# ip authentication keychain eigrp Test1 routeKeys	Associates a keychain with this EIGRP process for this interface. This configuration overrides the authentication configuration set in the router VRF mode. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
Step 10	ip authentication mode eigrp <i>instance-tag md5</i> Example: switch(config-if)# ip authentication mode eigrp Test1 md5	Configures the MD5 message digest authentication mode for this interface. This configuration overrides the authentication configuration set in the router VRF mode. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
Step 11	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure MD5 message digest authentication for EIGRP over Ethernet interface 1/2:

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config-router)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# ip router eigrp Test1
switch(config-if)# ip authentication keychain eigrp Test1 routeKeys
switch(config-if)# ip authentication mode eigrp Test1 md5
switch(config-if)# copy running-config startup-config
```

Configuring EIGRP Stub Routing

To configure a router for EIGRP stub routing, use the following command in address-family configuration mode:

Command	Purpose
<pre>switch(config-router-af)# stub [direct receive-only redistributed [direct] leak-map map-name]</pre> <p>Example:</p> <pre>switch(config-router-af)# eigrp stub redistributed</pre>	Configures a remote router as an EIGRP stub router. The map name can be any case-sensitive, alphanumeric string up to 20 characters.

This example shows how to configure a stub router to advertise directly connected and redistributed routes:

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# stub direct redistributed
switch(config-router-af)# copy running-config startup-config
```

Use the **show ip eigrp neighbor detail** command to verify that a router has been configured as a stub router. The last line of the output shows the stub status of the remote or spoke router. This example shows the output from the **show ip eigrp neighbor detail** command:

```
Router# show ip eigrp neighbor detail
IP-EIGRP neighbors for process 201
H   Address                Interface   Hold Uptime   SRTT    RTO  Q  Seq Type
                               (sec)          (ms)          Cnt  Num
0   10.1.1.2                Se3/1       11 00:00:59    1   4500  0   7
Version 12.1/1.2, Retrans: 2, Retries: 0
Stub Peer Advertising ( CONNECTED SUMMARY ) Routes
```

Configuring a Summary Address for EIGRP

You can configure a summary aggregate address for a specified interface. If any more specific routes are in the routing table, EIGRP will advertise the summary address out the interface with a metric equal to the minimum of all more specific routes. See the [“Route Summarization” section on page 6-6](#).

To configure a summary aggregate address, use the following command in interface configuration mode:

Command	Purpose
<pre>switch(config-if)# ip summary-address eigrp <i>instance-tag</i> <i>ip-prefix/length</i> [<i>distance</i> leak-map <i>map-name</i>]</pre> <p>Example:</p> <pre>switch(config-if)# ip summary-address eigrp Test1 192.0.2.0/8</pre>	<p>Configures a summary aggregate address as either an IP address and network mask, or an IP prefix/length. The instance tag and map name can be any case-sensitive, alphanumeric string up to 20 characters.</p> <p>You can optionally configure the administrative distance for this aggregate address. The default administrative distance is 5 for aggregate addresses.</p>

This example causes EIGRP to summarize network 192.0.2.0 out Ethernet 1/2 only:

```
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# ip summary-address eigrp Test1 192.0.2.0 255.255.255.0
```

Redistributing Routes into EIGRP

You can redistribute routes in EIGRP from other routing protocols.

BEFORE YOU BEGIN

Ensure that you have enabled the EIGRP feature (see the [“Enabling the EIGRP Feature”](#) section on page 6-9).

You must configure the metric (either through the default-metric configuration option or through a route map) for routes redistributed from any other protocol.

You must create a route map to control the types of routes that are redistributed into EIGRP. See [Chapter 16, “Configuring Route Policy Manager.”](#)

SUMMARY STEPS

1. **configure terminal**
2. **router eigrp** *instance-tag*
3. **address-family ipv4 unicast**
4. **redistribute** { *bgp as* | { **eigrp** | **ospf** | **ospfv3** | **rip** } *instance-tag* | **direct** | **static** } **route-map** *name*
5. **default-metric** *bandwidth delay reliability loading mtu*
6. **show ip eigrp route-map statistics redistribute**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router eigrp instance-tag Example: switch(config)# router eigrp Test1 switch(config-router)#	<p>Creates a new EIGRP process with the configured instance tag. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.</p> <p>If you configure an <i>instance-tag</i> that does not qualify as an AS number, you must use the autonomous-system command to configure the AS number explicitly or this EIGRP instance will remain in the shutdown state.</p>
Step 3	address-family ipv4 unicast Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	Enters the address-family configuration mode. This command is optional for IPv4.
Step 4	redistribute {bgp as {eigrp ospf ospfv3 rip} instance-tag direct static} route-map name Example: switch(config-router-af)# redistribute bgp 100 route-map BGPFilter	Injects routes from one routing domain into EIGRP. The instance tag and map name can be any case-sensitive, alphanumeric string up to 20 characters.
Step 5	default-metric bandwidth delay reliability loading mtu Example: switch(config-router-af)# default-metric 500000 30 200 1 1500	<p>Sets the metrics assigned to routes learned through route redistribution. The default values are as follows:</p> <ul style="list-style-type: none"> bandwidth—100000 Kb/s delay—100 (10 microsecond units) reliability—255 loading—1 MTU—1492
Step 6	show ip eigrp route-map statistics redistribute Example: switch(config-router-af)# show ip eigrp route-map statistics redistribute bgp	Displays information about EIGRP route map statistics.
Step 7	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to redistribute BGP into EIGRP for IPv4:

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config-router)# redistribute bgp 100 route-map BGPFilter
switch(config-router)# default-metric 500000 30 200 1 1500
switch(config-router)# copy running-config startup-config
```

Limiting the Number of Redistributed Routes

Route redistribution can add many routes to the EIGRP route table. You can configure a maximum limit to the number of routes accepted from external protocols. EIGRP provides the following options to configure redistributed route limits:

- **Fixed limit**—Logs a message when EIGRP reaches the configured maximum. EIGRP does not accept any more redistributed routes. You can optionally configure a threshold percentage of the maximum where EIGRP will log a warning when that threshold is passed.
- **Warning only**—Logs a warning only when EIGRP reaches the maximum. EIGRP continues to accept redistributed routes.
- **Withdraw**—Start the timeout period when EIGRP reaches the maximum. After the timeout period, EIGRP requests all redistributed routes if the current number of redistributed routes is less than the maximum limit. If the current number of redistributed routes is at the maximum limit, EIGRP withdraws all redistributed routes. You must clear this condition before EIGRP accepts more redistributed routes.

You can optionally configure the timeout period.

BEFORE YOU BEGIN

Ensure that you have enabled the EIGRP feature (see the [“Enabling the EIGRP Feature”](#) section on page 6-9).

SUMMARY STEPS

1. **configure terminal**
2. **router eigrp** *instance-tag*
3. **redistribute** {*bgp id* | *direct* | *eigrp id* | *ospf id* | *rip id* | *static*} **route-map** *map-name*
4. **redistribute maximum-prefix** *max* [*threshold*] [**warning-only** | **withdraw** [*num-retries* *timeout*]]
5. (Optional) **show running-config eigrp**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router eigrp instance-tag Example: switch(config)# router eigrp Test1 switch(config-router)#	Creates a new EIGRP instance with the configured instance tag.
Step 3	redistribute {bgp id direct eigrp id ospf id rip id static} route-map map-name Example: switch(config-router)# redistribute bgp route-map FilterExternalBGP	Redistributes the selected protocol into EIGRP through the configured route map.
Step 4	redistribute maximum-prefix max [threshold] [warning-only withdraw [num-retries timeout]] Example: switch(config-router)# redistribute maximum-prefix 1000 75 warning-only	Specifies a maximum number of prefixes that EIGRP will distribute. The range is from 0 to 65536. Optionally specifies the following: <ul style="list-style-type: none"> • threshold—Percent of maximum prefixes that will trigger a warning message. • warning-only—Logs an warning message when the maximum number of prefixes is exceeded. • withdraw—Withdraws all redistributed routes. Optionally tries to retrieve the redistributed routes. The <i>num-retries</i> range is from 1 to 12. The <i>timeout</i> is from 60 to 600 seconds. The default is 300 seconds. Use clear ip eigrp redistribution if all routes are withdrawn.
Step 5	show running-config eigrp Example: switch(config-router)# show running-config eigrp	(Optional) Displays the EIGRP configuration.
Step 6	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to limit the number of redistributed routes into EIGRP:

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config-router)# redistribute bgp route-map FilterExternalBGP
switch(config-router)# redistribute maximum-prefix 1000 75
```

Configuring Load Balancing in EIGRP

You can configure load balancing in EIGRP. You can configure the number of Equal Cost Multiple Path (ECMP) routes using the maximum paths option. See the [“Configuring Load Balancing in EIGRP” section on page 6-21](#).

BEFORE YOU BEGIN

Ensure that you have enabled the EIGRP feature (see the [“Enabling the EIGRP Feature” section on page 6-9](#)).

SUMMARY STEPS

1. **configure terminal**
2. **router eigrp** *instance-tag*
3. **address-family ipv4 unicast**
4. **maximum-paths** *num-paths*
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router eigrp <i>instance-tag</i> Example: switch(config)# router eigrp Test1 switch(config-router)#	<p>Creates a new EIGRP process with the configured instance tag. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.</p> <p>If you configure an <i>instance-tag</i> that does not qualify as an AS number, you must use the autonomous-system command to configure the AS number explicitly or this EIGRP instance will remain in the shutdown state.</p>
Step 3	address-family ipv4 unicast Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	Enters the address-family configuration mode. This command is optional for IPv4.

	Command	Purpose
Step 4	maximum-paths <i>num-paths</i> Example: switch(config-router-af)# maximum-paths 5	Sets the number of equal cost paths that EIGRP will accept in the route table. The range is from 1 to 16. The default is 8.
Step 5	copy running-config startup-config Example: switch(config-router-af)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure equal cost load balancing for EIGRP over IPv4 with a maximum of six equal cost paths:

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config-router)# maximum-paths 6
switch(config-router)# copy running-config startup-config
```

Adjusting the Interval Between Hello Packets and the Hold Time

You can adjust the interval between hello messages and the hold time.

By default, hello messages are sent every 5 seconds. The hold time is advertised in hello messages and indicates to neighbors the length of time that they should consider the sender valid. The default hold time is three times the hello interval, or 15 seconds.

To change the interval between hello packets, use the following command in interface configuration mode:

Command	Purpose
switch(config-if)# ip hello-interval eigrp <i>instance-tag seconds</i> Example: switch(config-if)# ip hello-interval eigrp Test1 30	Configures the hello interval for an EIGRP routing process. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters. The range is from 1 to 65535 seconds. The default is 5.

On very congested and large networks, the default hold time might not be sufficient time for all routers to receive hello packets from their neighbors. In this case, you might want to increase the hold time.

To change the hold time, use the following command in interface configuration mode:

Command	Purpose
switch(config-if)# ip hold-time eigrp <i>instance-tag seconds</i> Example: switch(config-if)# ip hold-time eigrp Test1 30	Configures the hold time for an EIGRP routing process. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters. The range is from 1 to 65535.

Use the **show ip eigrp interface detail** command to verify timer configuration.

Disabling Split Horizon

You can use split horizon to block route information from being advertised by a router out of any interface from which that information originated. Split horizon usually optimizes communications among multiple routing switches, particularly when links are broken.

By default, split horizon is enabled on all interfaces.

To disable split horizon, use the following command in interface configuration mode:

Command	Purpose
<pre>switch(config-if)# no ip split-horizon eigrp instance-tag</pre> <p>Example:</p> <pre>switch(config-if)# no ip split-horizon eigrp Test1</pre>	Disables split horizon.

Tuning EIGRP

You can configure optional parameters to tune EIGRP for your network.

You can configure the following optional parameters in address-family configuration mode:

Command	Purpose
<pre>default-information originate [always route-map map-name]</pre> <p>Example:</p> <pre>switch(config-router-af)# default-information originate always</pre>	Originates or accepts the default route with prefix 0.0.0.0/0. When a route map is supplied, the default route is originated only when the route map yields a true condition. The map name can be any case-sensitive, alphanumeric string up to 20 characters.
<pre>distance internal external</pre> <p>Example:</p> <pre>switch(config-router-af)# distance 25 100</pre>	Configures the administrative distance for this EIGRP process. The range is from 1 to 255. The internal value sets the distance for routes learned from within the same autonomous system (the default value is 90). The external value sets the distance for routes learned from an external autonomous system (the default value is 170).
<pre>metric maximum-hops hop-count</pre> <p>Example:</p> <pre>switch(config-router-af)# metric maximum-hops 70</pre>	Sets maximum allowed hops for an advertised route. Routes over this maximum are advertised as unreachable. The range is from 1 to 255. The default is 100.

Command	Purpose
metric weights <i>tos k1 k2 k3 k4 k5</i> Example: switch(config-router-af)# metric weights 0 1 3 2 1 0	Adjusts the EIGRP metric or K value. EIGRP uses the following formula to determine the total metric to the network: $\text{metric} = [k1 * \text{bandwidth} + (k2 * \text{bandwidth}) / (256 - \text{load}) + k3 * \text{delay}] * [k5 / (\text{reliability} + k4)]$ Default values and ranges are as follows: <ul style="list-style-type: none"> • TOS—0. The range is from 0 to 8. • k1—1. The range is from 0 to 255. • k2—0. The range is from 0 to 255. • k3—1. The range is from 0 to 255. • k4—0. The range is from 0 to 255. • k5—0. The range is from 0 to 255.
timers active-time { <i>time-limit</i> disabled } Example: switch(config-router-af)# timers active-time 200.	Sets the time the router waits in minutes (after sending a query) before declaring the route to be stuck in the active (SIA) state. The range is from 1 to 65535. The default is 3.

You can configure the following optional parameters in interface configuration mode:

Command	Purpose
ip bandwidth eigrp <i>instance-tag bandwidth</i> Example: switch(config-if)# ip bandwidth eigrp Test1 30000	Configures the bandwidth metric for EIGRP on an interface. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters. The bandwidth range is from 1 to 2,560,000,000 Kb/s.
ip bandwidth-percent eigrp <i>instance-tag percent</i> Example: switch(config-if)# ip bandwidth-percent eigrp Test1 30	Configures the percentage of bandwidth that EIGRP might use on an interface. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters. The percent range is from 0 to 100. The default is 50.
no ip delay eigrp <i>instance-tag delay</i> Example: switch(config-if)# ip delay eigrp Test1 100	Configures the delay metric for EIGRP on an interface. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters. The delay range is from 1 to 16777215 (in tens of microseconds).
ip distribute-list eigrp <i>instance-tag {prefix-list name route-map name} {in out}</i> Example: switch(config-if)# ip distribute-list eigrp Test1 route-map EigrpTest in	Configures the route filtering policy for EIGRP on this interface. The instance tag, prefix list name, and route map name can be any case-sensitive, alphanumeric string up to 20 characters.

Command	Purpose
no ip next-hop-self eigrp <i>instance-tag</i> Example: switch(config-if)# ip next-hop-self eigrp Test1	Configures EIGRP to use the received next-hop address rather than the address for this interface. The default is to use the IP address of this interface for the next-hop address. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
ip offset-list eigrp <i>instance-tag</i> { prefix-list <i>name</i> route-map <i>name</i> } { in out } <i>offset</i> Example: switch(config-if)# ip offset-list eigrp Test1 prefix-list EigrpList in	Adds an offset to incoming and outgoing metrics to routes learned by EIGRP. The instance tag, prefix list name, and route map name can be any case-sensitive, alphanumeric string up to 20 characters.
ip passive-interface eigrp <i>instance-tag</i> Example: switch(config-if)# ip passive-interface eigrp Test1	Suppresses EIGRP hellos, which prevents neighbors from forming and sending routing updates on an EIGRP interface. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.

Configuring Virtualization for EIGRP

You can create multiple VRFs and use the same or multiple EIGRP processes in each VRF. You assign an interface to a VRF.



Note

Configure all other parameters for an interface after you configure the VRF for an interface. Configuring a VRF for an interface deletes all other configuration for that interface.

BEFORE YOU BEGIN

Ensure that you have enabled the EIGRP feature (see the [“Enabling the EIGRP Feature”](#) section on page 6-9).

Create the VRFs.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **router eigrp** *instance-tag*
4. **interface ethernet** *slot/port*
5. **no switchport**
6. **vrf member** *vrf-name*
7. **ip router eigrp** *instance-tag*
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	vrf context <i>vrf-name</i> Example: switch(config)# vrf context RemoteOfficeVRF switch(config-vrf)#	Creates a new VRF and enters VRF configuration mode. The VRN name can be any case-sensitive, alphanumeric string up to 20 characters.
Step 3	router eigrp <i>instance-tag</i> Example: switch(config)# router eigrp Test1 switch(config-router)#	<p>Creates a new EIGRP process with the configured instance tag. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.</p> <p>If you configure an <i>instance-tag</i> that does not qualify as an AS number, you must use the autonomous-system command to configure the AS number explicitly or this EIGRP instance will remain in the shutdown state.</p>
Step 4	interface ethernet <i>slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode. Use ? to find the slot and port ranges.
Step 5	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 6	vrf member <i>vrf-name</i> Example: switch(config-if)# vrf member RemoteOfficeVRF	Adds this interface to a VRF. The VRF name can be any case-sensitive, alphanumeric string up to 20 characters.
Step 7	ip router eigrp <i>instance-tag</i> Example: switch(config-if)# ip router eigrp Test1	Adds this interface to the EIGRP process. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
Step 8	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to create a VRF and add an interface to the VRF:

```
switch# configure terminal
switch(config)# vrf context NewVRF
switch(config-vrf)# router eigrp Test1
switch(config-router)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# ip router eigrp Test1
switch(config-if)# vrf member NewVRF
switch(config-if)# copy running-config startup-config
```

Verifying the EIGRP Configuration

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

Command	Purpose
show ip eigrp [<i>instance-tag</i>]	Displays a summary of the configured EIGRP processes.
show ip eigrp [<i>instance-tag</i>] interfaces [<i>type number</i>] [brief] [detail]	Displays information about all configured EIGRP interfaces.
show ip eigrp <i>instance-tag</i> neighbors [<i>type number</i>]	Displays information about all the EIGRP neighbors. Use this command to verify the EIGRP neighbor configuration.
show ip eigrp [<i>instance-tag</i>] route [<i>ip-prefix/length</i>] [active] [all-links] [detail-links] [pending] [summary] [zero-successors] [vrf <i>vrf-name</i>]	Displays information about all the EIGRP routes.
show ip eigrp [<i>instance-tag</i>] topology [<i>ip-prefix/length</i>] [active] [all-links] [detail-links] [pending] [summary] [zero-successors] [vrf <i>vrf-name</i>]	Displays information about the EIGRP topology table.
show running-configuration eigrp	Displays the current running EIGRP configuration.

Displaying EIGRP Statistics

To display EIGRP statistics, use the following commands:

Command	Purpose
show ip eigrp [<i>instance-tag</i>] accounting [vrf <i>vrf-name</i>]	Displays accounting statistics for EIGRP.
show ip eigrp [<i>instance-tag</i>] route-map statistics redistribute	Displays redistribution statistics for EIGRP.
show ip eigrp [<i>instance-tag</i>] traffic [vrf <i>vrf-name</i>]	Displays traffic statistics for EIGRP.

Configuration Examples for EIGRP

This example shows how to configure EIGRP:

```
feature eigrp
interface ethernet 1/2
 no switchport
 ip address 192.0.2.55/24
 ip router eigrp Test1
 no shutdown
router eigrp Test1
 router-id 192.0.2.1
```

Related Topics

See [Chapter 16, “Configuring Route Policy Manager”](#) for more information on route maps.

Additional References

For additional information related to implementing EIGRP, see the following sections:

- [Related Documents, page 6-28](#)
- [MIBs, page 6-28](#)

Related Documents

Related Topic	Document Title
EIGRP CLI commands	<i>Cisco Nexus 3000 Series Command Reference</i>
http://www.cisco.com/warp/public/103/1.html	<i>Introduction to EIGRP Tech Note</i>
http://www.cisco.com/en/US/tech/tk365/technologies_q_and_a_item09186a008012dac4.shtml	EIGRP Frequently Asked Questions

MIBs

MIBs	MIBs Link
CISCO-EIGRP-MIB	To locate and download MIBs, go to the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml



Configuring Basic BGP

This chapter describes how to configure Border Gateway Protocol (BGP) on a Cisco NX-OS switch.

This chapter includes the following sections:

- [Information About Basic BGP, page 7-1](#)
- [Licensing Requirements for Basic BGP, page 7-7](#)
- [Prerequisites for Basic BGP, page 7-7](#)
- [Guidelines and Limitations for BGP, page 7-8](#)
- [CLI Configuration Modes, page 7-8](#)
- [Default Settings, page 7-10](#)
- [Configuring Basic BGP, page 7-10](#)
- [Verifying the Basic BGP Configuration, page 7-25](#)
- [Displaying BGP Statistics, page 7-27](#)
- [Configuration Examples for Basic BGP, page 7-27](#)
- [Related Topics, page 7-27](#)
- [Where to Go Next, page 7-27](#)
- [Additional References, page 7-28](#)

Information About Basic BGP

Cisco NX-OS supports BGP version 4, which includes multiprotocol extensions that allow BGP to carry routing information and multiple Layer 3 protocol address families. BGP uses TCP as a reliable transport protocol to create TCP sessions with other BGP-enabled switches.

BGP uses a path-vector routing algorithm to exchange routing information between BGP-enabled networking switches or *BGP speakers*. Based on this information, each BGP speaker determines a path to reach a particular destination while detecting and avoiding paths with routing loops. The routing information includes the actual route prefix for a destination, the path of autonomous systems to the destination, and additional path attributes.

BGP selects a single path, by default, as the best path to a destination host or network. Each path carries well-known mandatory, well-known discretionary, and optional transitive attributes that are used in BGP best-path analysis. You can influence BGP path selection by altering some of these attributes by configuring BGP policies. See the [“Route Policies and Resetting BGP Sessions”](#) section on page 8-3 for more information.

BGP also supports load balancing or equal-cost multipath (ECMP). See the [“Load Sharing and Multipath”](#) section on page 8-7 for more information.

This section includes the following topics:

- [BGP Autonomous Systems](#), page 7-2
- [Administrative Distance](#), page 7-2
- [BGP Peers](#), page 7-3
- [BGP Router Identifier](#), page 7-3
- [BGP Path Selection](#), page 7-4
- [BGP and the Unicast RIB](#), page 7-7
- [BGP Virtualization](#), page 7-7

BGP Autonomous Systems

An *autonomous system* (AS) is a network controlled by a single administration entity. An autonomous system forms a routing domain with one or more interior gateway protocols (IGPs) and a consistent set of routing policies. BGP supports 16-bit and 32-bit autonomous system numbers. For more information, see the [“Autonomous Systems”](#) section on page 1-5.

Separate BGP autonomous systems dynamically exchange routing information through external BGP (eBGP) peering sessions. BGP speakers within the same autonomous system can exchange routing information through internal BGP (iBGP) peering sessions.

4-Byte AS Number Support

BGP supports 2-byte or 4-byte AS numbers. Cisco NX-OS displays 4-byte AS numbers in plain-text notation (that is, as 32-bit integers). You can configure 4-byte AS numbers as either plain-text notation (for example, 1 to 4294967295), or AS.dot notation (for example, 1.0). For more information, see the [“Autonomous Systems”](#) section on page 1-5.

Administrative Distance

An *administrative distance* is a rating of the trustworthiness of a routing information source. By default, BGP uses the administrative distances shown in [Table 7-1](#).

Table 7-1 BGP Default Administrative Distances

Distance	Default Value	Function
External	20	Applied to routes learned from eBGP.
Internal	200	Applied to routes learned from iBGP.
Local	200	Applied to routes originated by the router.



Note

The administrative distance does not influence the BGP path selection algorithm, but it does influence whether BGP-learned routes are installed in the IP routing table.

For more information, see the [“Administrative Distance”](#) section on page 1-7.

BGP Peers

A BGP speaker does not discover another BGP speaker automatically. You must configure the relationships between BGP speakers. A *BGP peer* is a BGP speaker that has an active TCP connection to another BGP speaker.

BGP Sessions

BGP uses TCP port 179 to create a TCP session with a peer. When a TCP connection is established between peers, each BGP peer initially exchanges all of its routes—the complete BGP routing table—with the other peer. After this initial exchange, the BGP peers send only incremental updates when a topology change occurs in the network or when a routing policy change occurs. In the periods of inactivity between these updates, peers exchange special messages called *keepalives*. The *hold time* is the maximum time limit that can elapse between receiving consecutive BGP update or keepalive messages.

Cisco NX-OS supports the following peer configuration options:

- Individual IPv4 or IPv6 address—BGP establishes a session with the BGP speaker that matches the remote address and AS number.
- IPv4 or IPv6 prefix peers for a single AS number—BGP establishes sessions with BGP speakers that match the prefix and the AS number.
- Dynamic AS number prefix peers—BGP establishes sessions with BGP speakers that match the prefix and an AS number from a list of configured AS numbers.

Dynamic AS Numbers for Prefix Peers

Cisco NX-OS accepts a range or list of AS numbers to establish BGP sessions. For example, if you configure BGP to use IPv4 prefix 192.0.2.0/8 and AS numbers 33, 66, and 99, BGP establishes a session with 192.0.2.1 with AS number 66 but rejects a session from 192.0.2.2 with AS number 50.)

Cisco NX-OS does not associate prefix peers with dynamic AS numbers as either interior BGP (iBGP) or external BGP (eBGP) sessions until after the session is established. See [Chapter 8, “Configuring Advanced BGP,”](#) for more information on iBGP and eBGP.



Note

The dynamic AS number prefix peer configuration overrides the individual AS number configuration that is inherited from a BGP template. See [Chapter 8, “Configuring Advanced BGP,”](#) for more information on templates.

BGP Router Identifier

To establish BGP sessions between peers, BGP must have a *router ID*, which is sent to BGP peers in the OPEN message when a BGP session is established. The BGP router ID is a 32-bit value that is often represented by an IPv4 address. You can configure the router ID. By default, Cisco NX-OS sets the router ID to the IPv4 address of a loopback interface on the router. If no loopback interface is configured

on the router, then the software chooses the highest IPv4 address configured to a physical interface on the router to represent the BGP router ID. The BGP router ID must be unique to the BGP peers in a network.

If BGP does not have a router ID, it cannot establish any peering sessions with BGP peers.

BGP Path Selection

Although BGP might receive advertisements for the same route from multiple sources, BGP selects only one path as the best path. BGP puts the selected path in the IP routing table and propagates the path to its peers.

The best-path algorithm runs each time that a path is added or withdrawn for a given network. The best-path algorithm also runs if you change the BGP configuration. BGP selects the best path from the set of valid paths available for a given network.

Cisco NX-OS implements the BGP best-path algorithm in the following steps:

-
- | | |
|---------------|--|
| Step 1 | Compares two paths to determine which is better (see the “Step 1—Comparing Pairs of Paths” section on page 7-4). |
| Step 2 | Iterates over all paths and determines in which order to compare the paths to select the overall best path (see the “Step 2—Determining the Order of Comparisons” section on page 7-6). |
| Step 3 | Determines whether the old and new best paths differ enough so that the new best path should be used (see the “Step 3—Determining the Best-Path Change Suppression” section on page 7-6). |
-

**Note**

The order of comparison determined in Part 2 is important. Consider the case where you have three paths, A, B, and C. When Cisco NX-OS compares A and B, it chooses A. When Cisco NX-OS compares B and C, it chooses B. But when Cisco NX-OS compares A and C, it might not choose A because some BGP metrics apply only among paths from the same neighboring autonomous system and not among all paths.

The path selection uses the BGP AS-path attribute. The AS-path attribute includes the list of autonomous system numbers (AS numbers) traversed in the advertised path. If you subdivide your BGP autonomous system into a collection or confederation of autonomous systems, the AS path contains confederation segments that list these locally defined autonomous systems.

Step 1—Comparing Pairs of Paths

This first step in the BGP best-path algorithm compares two paths to determine which path is better. The following sequence describes the basic steps that Cisco NX-OS uses to compare two paths to determine the better path:

1. Cisco NX-OS chooses a valid path for comparison. (For example, a path that has an unreachable next-hop is not valid.)
2. Cisco NX-OS chooses the path with the highest weight.
3. Cisco NX-OS chooses the path with the highest local preference.
4. If one of the paths is locally originated, Cisco NX-OS chooses that path.

5. Cisco NX-OS chooses the path with the shorter AS path.



Note When calculating the length of the AS path, Cisco NX-OS ignores confederation segments, and counts AS sets as 1. See the [“AS Confederations” section on page 8-5](#) for more information.

6. Cisco NX-OS chooses the path with the lower origin. The Interior Gateway Protocol (IGP) is considered lower than EGP.
7. Cisco NX-OS chooses the path with the lower multi-exit discriminator (MED).

You can configure a number of options that affect whether or not this step is performed. In general, Cisco NX-OS compares the MED of both paths if the paths were received from peers in the same autonomous system; otherwise, Cisco NX-OS skips the MED comparison.

You can configure Cisco NX-OS to always perform the best-path algorithm MED comparison, regardless of the peer autonomous system in the paths. See the [“Tuning the Best-Path Algorithm” section on page 8-10](#) for more information. Otherwise, Cisco NX-OS will perform a MED comparison that depends on the AS-path attributes of the two paths being compared:

- a. If a path has no AS path or the AS path starts with an AS_SET, then the path is internal, and Cisco NX-OS compares the MED to other internal paths.
- b. If the AS path starts with an AS_SEQUENCE, then the peer autonomous system is the first AS number in the sequence, and Cisco NX-OS compares the MED to other paths that have the same peer autonomous system.
- c. If the AS path contains only confederation segments or starts with confederation segments followed by an AS_SET, the path is internal and Cisco NX-OS compares the MED to other internal paths.
- d. If the AS path starts with confederation segments followed by an AS_SEQUENCE, then the peer autonomous system is the first AS number in the AS_SEQUENCE, and Cisco NX-OS compares the MED to other paths that have the same peer autonomous system.



Note If Cisco NX-OS receives no MED attribute with the path, then Cisco NX-OS considers the MED to be 0 unless you configure the best-path algorithm to set a missing MED to the highest possible value. See the [“Tuning the Best-Path Algorithm” section on page 8-10](#) for more information.

- e. If the nondeterministic MED comparison feature is enabled, the best path algorithm uses the Cisco IOS style of MED comparison. See the [“Tuning the Best-Path Algorithm” section on page 8-10](#) for more information.
8. If one path is from an internal peer and the other path is from an external peer, then Cisco NX-OS chooses the path from the external peer.
9. If the paths have different IGP metrics to their next-hop addresses, then Cisco NX-OS chooses the path with the lower IGP metric.
10. Cisco NX-OS uses the path that was selected by the best-path algorithm the last time that it was run.

If all path parameters in Step 1 through Step 9 are the same, then you can configure the best-path algorithm to compare the router IDs. See the [“Tuning the Best-Path Algorithm” section on page 8-10](#) for more information. If the path includes an originator attribute, then Cisco NX-OS uses that attribute as the router ID to compare to; otherwise, Cisco NX-OS uses the router ID of the peer that sent the path. If the paths have different router IDs, Cisco NX-OS chooses the path with the lower router ID.

**Note**

When using the attribute originator as the router ID, it is possible that two paths have the same router ID. It is also possible to have two BGP sessions with the same peer router, and therefore you can receive two paths with the same router ID.

11. Cisco NX-OS selects the path with the shorter cluster length. If a path was not received with a cluster list attribute, the cluster length is 0.
12. Cisco NX-OS chooses the path received from the peer with the lower IP address. Locally generated paths (for example, redistributed paths) have a peer IP address of 0.

**Note**

Paths that are equal after step 9 can be used for multipath if you configure multipath. See the [“Load Sharing and Multipath” section on page 8-7](#) for more information.

Step 2—Determining the Order of Comparisons

The second step of the BGP best-path algorithm implementation is to determine the order in which Cisco NX-OS compares the paths:

1. Cisco NX-OS partitions the paths into groups. Within each group Cisco NX-OS compares the MED among all paths. Cisco NX-OS uses the same rules as in the [“Step 1—Comparing Pairs of Paths” section on page 7-4](#) to determine whether MED can be compared between any two paths. Typically, this comparison results in one group being chosen for each neighbor autonomous system. If you configure the **bgp bestpath med always** command, then Cisco NX-OS chooses just one group that contains all the paths.
2. Cisco NX-OS determines the best path in each group by iterating through all paths in the group and keeping track of the best one so far. Cisco NX-OS compares each path with the temporary best path found so far and if the new path is better, it becomes the new temporary best path and Cisco NX-OS compares it with the next path in the group.
3. Cisco NX-OS forms a set of paths that contain the best path selected from each group in Step 2. Cisco NX-OS selects the overall best path from this set of paths by going through them as in Step 2.

Step 3—Determining the Best-Path Change Suppression

The next part of the implementation is to determine whether Cisco NX-OS will use the new best path or suppress the new best path. The router can continue to use the existing best path if the new one is identical to the old path (if the router ID is the same). Cisco NX-OS continues to use the existing best path to avoid route changes in the network.

You can turn off the suppression feature by configuring the best-path algorithm to compare the router IDs. See the [“Tuning the Best-Path Algorithm” section on page 8-10](#) for more information. If you configure this feature, the new best path is always preferred to the existing one.

You cannot suppress the best-path change if any of the following conditions occur:

- The existing best path is no longer valid.
- Either the existing or new best paths were received from internal (or confederation) peers or were locally generated (for example, by redistribution).
- The paths were received from the same peer (the paths have the same router ID).
- The paths have different weights, local preferences, origins, or IGP metrics to their next-hop addresses.

- The paths have different MEDs.

BGP and the Unicast RIB

BGP communicates with the Unicast Routing Information Base (unicast RIB) to store IPv4 routes in the unicast routing table. After selecting the best path, if BGP determines that the best path change needs to be reflected in the routing table, it sends a route update to the unicast RIB.

BGP receives route notifications regarding changes to its routes in the unicast RIB. It also receives route notifications about other protocol routes to support redistribution.

BGP also receives notifications from the unicast RIB regarding next-hop changes. BGP uses these notifications to keep track of the reachability and IGP metric to the next-hop addresses.

Whenever the next-hop reachability or IGP metrics in the unicast RIB change, BGP triggers a best-path recalculation for affected routes.

BGP Virtualization

BGP supports virtual routing and forwarding (VRF) instances. By default, Cisco NX-OS places you in the default VRF unless you specifically configure another VRF. For more information, see [Chapter 14, “Configuring Layer 3 Virtualization.”](#)

Licensing Requirements for Basic BGP

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	BGP requires a LAN Enterprise Services license. For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the <i>Cisco NX-OS Licensing Guide</i> .
	Note Make sure the LAN Base Services license is installed on the switch to enable Layer 3 interfaces.

Prerequisites for Basic BGP

BGP has the following prerequisites:

- You must enable the BGP feature (see the [“Enabling the BGP Feature”](#) section on page 7-11).
- You should have a valid router ID configured on the system.
- You must have an AS number, either assigned by a Regional Internet Registry (RIR) or locally administered.
- You must configure at least one IGP that is capable of recursive next-hop resolution.
- You must configure an address family under a neighbor for the BGP session establishment.

Guidelines and Limitations for BGP

BGP has the following configuration guidelines and limitations:

- The dynamic AS number prefix peer configuration overrides individual AS number configuration inherited from a BGP template.
- If you configure a dynamic AS number for prefix peers in an AS confederation, BGP establishes sessions with only the AS numbers in the local confederation.
- BGP sessions created through a dynamic AS number prefix peer ignore any configured external BGP (eBGP) multihop time-to-live (TTL) value or a disabled check for directly connected peers.
- You must configure a router ID for BGP to avoid automatic router ID changes and session flaps.
- You must use the maximum-prefix configuration option per peer to restrict the number of routes received and system resources used.
- You must configure the update source to establish a session with BGP/eBGP multihop sessions.
- You must specify a BGP policy if you configure redistribution.
- You must define the BGP router ID within a VRF.
- If you decrease the keepalive and hold timer values, you might experience BGP session flaps.
- If you configure VRFs, enter the desired VRF (see [Chapter 14, “Configuring Layer 3 Virtualization”](#)).

CLI Configuration Modes

The following sections describe how to enter each of the CLI configuration modes for BGP. From a mode, you can enter the ? command to display the commands available in that mode.

This section includes the following topics:

- [Global Configuration Mode, page 7-8](#)
- [Address Family Configuration Mode, page 7-9](#)
- [Neighbor Configuration Mode, page 7-9](#)
- [Neighbor Address Family Configuration Mode, page 7-9](#)

Global Configuration Mode

Use global configuration mode to create a BGP process and configure advanced features such as AS confederation and route dampening. For more information, see [Chapter 8, “Configuring Advanced BGP”](#).

This example shows how to enter router configuration mode:

```
switch# configuration
switch(config)# router bgp 64496
switch(config-router)#
```

BGP supports virtual routing and forwarding (VRF). You can configure BGP within the appropriate VRF if you are using VRFs in your network. See the [“Configuring Virtualization” section on page 8-46](#) for more information.

This example shows how to enter VRF configuration mode:

```
switch(config)# router bgp 64497  
switch(config-router)# vrf vrf_A  
switch(config-router-vrf)#
```

Address Family Configuration Mode

You can optionally configure the address families that BGP supports. Use the **address-family** command in router configuration mode to configure features for an address family. Use the **address-family** command in neighbor configuration mode to configure the specific address family for the neighbor.

You must configure the address families if you are using route redistribution, address aggregation, load balancing, and other advanced features.

This example shows how to enter address family configuration mode from the router configuration mode:

```
switch(config)# router bgp 64496  
switch(config-router)# address-family ipv6 unicast  
switch(config-router-af)#
```

This example shows how to enter VRF address family configuration mode if you are using VRFs:

```
switch(config)# router bgp 64497  
switch(config-router)# vrf vrf_A  
switch(config-router-vrf)# address-family ipv6 unicast  
switch(config-router-vrf-af)#
```

Neighbor Configuration Mode

Cisco NX-OS provides the neighbor configuration mode to configure BGP peers. You can use neighbor configuration mode to configure all parameters for a peer.

This example shows how to enter neighbor configuration mode:

```
switch(config)# router bgp 64496  
switch(config-router)# neighbor 192.0.2.1  
switch(config-router-neighbor)#
```

This example shows how to enter VRF neighbor configuration mode:

```
switch(config)# router bgp 64497  
switch(config-router)# vrf vrf_A  
switch(config-router-vrf)# neighbor 192.0.2.1  
switch(config-router-vrf-neighbor)#
```

Neighbor Address Family Configuration Mode

An address family configuration submode inside the neighbor configuration submode is available for entering address family-specific neighbor configuration and enabling the address family for the neighbor. Use this mode for advanced features such as limiting the number of prefixes allowed for this neighbor and removing private AS numbers for eBGP.

With the introduction of RFC 5549 in Cisco NX-OS Release 6.0(2)U4(1), you can configure an IPv4 address family for a neighbor with an IPv6 address.

This example shows how to enter the IPv4 neighbor address family configuration mode for a neighbor with an IPv4 address:

```
switch(config)# router bgp 64496
switch(config-router)# neighbor 192.0.2.1
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)#
```

This example shows how to enter the IPv4 neighbor address family configuration mode for a neighbor with an IPv6 address:

```
switch(config)# router bgp 64496
switch(config-router)# neighbor 2001:db8::/64 eui64
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)#
```

This example shows how to enter the VRF IPv4 neighbor address family configuration mode for a neighbor with an IPv4 address:

```
switch(config)# router bgp 64497
switch(config-router)# vrf vrf_A
switch(config-router-vrf)# neighbor 209.165.201.1
switch(config-router-vrf-neighbor)# address-family ipv4 unicast
switch(config-router-vrf-neighbor-af)#
```

This example shows how to enter the VRF IPv4 neighbor address family configuration mode for a neighbor with an IPv6 address:

```
switch(config)# router bgp 64497
switch(config-router)# vrf vrf_A
switch(config-router-vrf)# neighbor 2001:db8::/64 eui64
switch(config-router-vrf-neighbor)# address-family ipv4 unicast
switch(config-router-vrf-neighbor-af)#
```

Default Settings

Table 7-2 lists the default settings for BGP parameters.

Table 7-2 Default BGP Parameters

Parameters	Default
BGP feature	Disabled
keep alive interval	60 seconds
hold timer	180 seconds

Configuring Basic BGP

To configure a basic BGP, you need to enable BGP and configure a BGP peer. Configuring a basic BGP network consists of a few required tasks and many optional tasks. You must configure a BGP routing process and BGP peers.

This section includes the following topics:

- [Enabling the BGP Feature, page 7-11](#)

- [Creating a BGP Instance, page 7-12](#)
- [Restarting a BGP Instance, page 7-13](#)
- [Shutting Down BGP, page 7-14](#)
- [Configuring BGP Peers, page 7-14](#)
- [Distributing the Default Static Route to All BGP VRFs, page 7-16](#)
- [Configuring Update Announcement Delay Timers, page 7-17](#)
- [Configuring BGP Reconnect Interval, page 7-19](#)
- [Configuring Dynamic AS Numbers for Prefix Peers, page 7-20](#)
- [Clearing BGP Information, page 7-22](#)



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 the BGP Feature

You must enable the BGP feature before you can configure BGP.

SUMMARY STEPS

1. **configure terminal**
2. **feature bgp**
3. (Optional) **show feature**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	feature bgp Example: switch(config)# feature bgp	Enables the BGP feature.
Step 3	show feature Example: switch(config)# show feature	(Optional) Displays enabled and disabled features.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

Use the **no feature bgp** command to disable the BGP feature and remove all associated configuration.

Command	Purpose
no feature bgp Example: switch(config)# no feature bgp	Disables the BGP feature and removes all associated configuration.

Creating a BGP Instance

You can create a BGP instance and assign a router ID to the BGP instance. See the [“BGP Router Identifier” section on page 7-3](#). Cisco NX-OS supports 2-byte or 4-byte autonomous system (AS) numbers in plain-text notation or as.dot notation. See the [“4-Byte AS Number Support” section on page 7-2](#) for more information.

BEFORE YOU BEGIN

Ensure that you have enabled the BGP feature (see the [“Enabling the BGP Feature” section on page 7-11](#)).

BGP must be able to obtain a router ID (for example, a configured loopback address).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. (Optional) **router-id** *ip-address*
4. (Optional) **address-family ipv4 unicast**
5. (Optional) **network ip-prefix** [**route-map** *map-name*]
6. (Optional) **show bgp all**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 64496 switch(config-router)#	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.

	Command	Purpose
Step 3	router-id <i>ip-address</i> Example: switch(config-router)# router-id 192.0.2.255	(Optional) Configures the BGP router ID. This IP address identifies this BGP speaker. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 4	address-family ipv4 unicast Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	(Optional) Enters global address family configuration mode for the specified address family. This command triggers an automatic notification and session reset for all BGP neighbors.
Step 5	network <i>ip-prefix</i> [route-map <i>map-name</i>] Example: switch(config-router-af)# network 192.0.2.0	(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table. For exterior protocols, the network command controls which networks are advertised. Interior protocols use the network command to determine where to send updates.
Step 6	show bgp all Example: switch(config-router-af)# show bgp all	(Optional) Displays information about all BGP address families.
Step 7	copy running-config startup-config Example: switch(config-router-af)# copy running-config startup-config	(Optional) Saves this configuration change.

Use the **no router bgp** command to remove the BGP process and the associated configuration.

Command	Purpose
no router bgp <i>autonomous-system-number</i> Example: switch(config)# no router bgp 201	Deletes the BGP process and the associated configuration.

This example shows how to enable BGP with the IPv4 unicast address family and manually add one network to advertise:

```
switch# configure terminal
switch(config)# router bgp 64496
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# network 192.0.2.0
switch(config-router-af)# copy running-config startup-config
```

Restarting a BGP Instance

You can restart a BGP instance and clear all peer sessions for the instance.

To restart a BGP instance and remove all associated peers, use the following command:

Command	Purpose
restart <i>bgp</i> <i>instance-tag</i>	Restarts the BGP instance and resets or reestablishes all peering sessions.
Example: switch(config)# restart bgp 201	

Shutting Down BGP

You can shut down the BGP protocol and gracefully disable BGP and retain the configuration.

To shut down BGP, use the following command in router configuration mode:

Command	Purpose
shutdown	Gracefully shuts down BGP.
Example: switch(config-router)# shutdown	

Configuring BGP Peers

You can configure a BGP peer within a BGP process. Each BGP peer has an associated keepalive timer and hold timers. You can set these timers either globally or for each BGP peer. A peer configuration overrides a global configuration.



Note

You must configure the address family under neighbor configuration mode for each peer.

BEFORE YOU BEGIN

Ensure that you have enabled the BGP feature (see the [“Enabling the BGP Feature”](#) section on page 7-11).

SUMMARY STEPS

1. **configure terminal**
2. **router** *bgp* *autonomous-system-number*
3. **neighbor** *ip-address* {*ipv4* | *ipv6*} **remote-as** *as-number*
4. (Optional) **description** *text*
5. (Optional) **timers** *keepalive-time* *hold-time*
6. (Optional) **shutdown**
7. **address-family** *ipv4* **unicast**
8. (Optional) **show** *bgp* *ipv4* **unicast** **neighbors**
9. (Optional) **copy** *running-config* *startup-config*

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 64496 switch(config-router)#	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.
Step 3	neighbor <i>ip-address</i> { ipv4 ipv6 } remote-as <i>as-number</i> Example: switch(config-router)# neighbor 209.165.201.1 remote-as 64497 switch(config-router-neighbor)#	Configures the specified address type and AS number for a remote BGP peer. The <i>ip-address</i> format is x.x.x.x. The IPv6 <i>address-format</i> is A:B::C:D.
Step 4	description <i>text</i> Example: switch(config-router-neighbor)# description Peer Router B switch(config-router-neighbor)#	(Optional) Adds a description for the neighbor. The description is an alphanumeric string up to 80 characters.
Step 5	timers <i>keepalive-time hold-time</i> Example: switch(config-router-neighbor)# timers 30 90	(Optional) Adds the keepalive and hold time BGP timer values for the neighbor. The range is from 0 to 3600 seconds. The default is 60 seconds for the keepalive time and 180 seconds for the hold time.
Step 6	shutdown Example: switch(config-router-neighbor)# shutdown	(Optional) Administratively shuts down this BGP neighbor. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 7	address-family ipv4 unicast Example: switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#	Enters neighbor address family configuration mode for the unicast specified address family.
Step 8	show bgp ipv4 unicast neighbors Example: switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors	(Optional) Displays information about BGP peers.
Step 9	copy running-config startup-config Example: switch(config-router-neighbor-af) copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure a BGP peer:

```
switch# configure terminal
switch(config)# router bgp 64496
switch(config-router)# neighbor 192.0.2.1 remote-as 64497
switch(config-router-neighbor)# description Peer Router B
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# copy running-config startup-config
```

Distributing the Default Static Route to All BGP VRFs

A new CLI **default-information originate** is added to distribute the default static route from non-default VRF to all BGP (VRF) Virtual Router Context and to install that route in the local BGP route table for all the BGP VRFs.

BEFORE YOU BEGIN

Ensure that you have enabled the BGP feature.

SUMMARY STEPS

- 1. **configure terminal**
- 2. **router bgp** *autonomous-system-number*
- 3. **vrf** *vrf-name*
- 4. **address-family ipv4 unicast**
- 5. **redistribute static route-map test**
- 6. **default-information originate**
- 7. **exit**
- 8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 64496 switch(config-router)#	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.
Step 3	vrf <i>vrf-name</i>	Configures the VRF name.

	Command	Purpose
Step 4	address-family ipv4 unicast Example: switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#	Enters global address family configuration mode for the specified address family. This command triggers an automatic notification and session reset for all BGP neighbors.
Step 5	redistribute static route-map test	Redistributes the static route map.
Step 6	default-information originate	Distributes the default static route from non-default VRF to all BGP (VRF) Virtual Router Context and to install that route in the local BGP route table for all the BGP VRFs.
Step 7	exit	Exits the configuration.
Step 8	copy running-config startup-config	Saves this configuration change.

This example shows how to distribute the default static route from non-default VRF to all BGP (VRF) Virtual Router Context and to install that route in the local BGP route table for all the BGP VRFs:

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# vrf green
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-vrf-af)# redistribute static route-map test
switch(config-router-vrf-af)# default-information originate
switch(config-router-vrf-af)# exit
switch(config)# copy running-config startup-config
```

This example shows how to configure default-route leaking from non-default VRF Green to another non-default VRF Shared:

```
router bgp 100
address-family ipv4 unicast
    redistribute static route-map test
    vrf Green
address-family ipv4 unicast
    redistribute static route-map test
    default-information originate
    vrf Shared
address-family ipv4 unicast
    redistribute static route-map test
```

Configuring Update Announcement Delay Timers

You can configure a minimum route advertisement interval (MRAI) between the sending of BGP routing updates by using the **advertisement-interval** command. Instead of announcing route updates as soon as they are triggered, BGP waits for the advertisement interval to expire before sending out the update. If there are other changes that happen during this interval, BGP can announce all these changes efficiently.

This delay timer is configured only for new route announcements and not for route withdrawal announcements. Route withdrawals must be announced immediately.


BEFORE YOU BEGIN

Ensure that you have enabled the BGP feature.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor** *prefix* **remote-as** **route-map** *map-name*
4. **address-family** **ipv4 unicast**
5. **advertisement-interval** *seconds*

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 64496 switch(config-router)#	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.
Step 3	neighbor <i>prefix</i> remote-as route-map <i>map-name</i> Example: switch(config-router)# neighbor 192.0.2.0/8 remote-as routemap BGPPeers switch(config-router-neighbor)#	Configures the IPv4 prefix and a route map for the list of accepted AS numbers for the remote BGP peers. The <i>prefix</i> format for IPv4 is x.x.x.x/length. The length range is from 1 to 32. The <i>map-name</i> can be any case-sensitive, alphanumeric string up to 63 characters.
Step 4	address-family ipv4 unicast Example: switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#	Enters global address family configuration mode for the specified address family. This command triggers an automatic notification and session reset for all BGP neighbors.
Step 5	advertisement-interval <i>seconds</i> Example: switch(config-router-neighbor-af)# advertisement-interval 300	Configures an interval that delays the announcement of new route updates.  Note This interval does not apply to route withdrawal announcements, which must be made immediately. The interval can range between 1 and 600 seconds.

This example shows how to configure an update announcement delay timer:

```
switch# configure terminal
switch(config)# router bgp 64496
switch(config-router)# neighbor 192.0.2.0/8 remote-as route-map BGPPeers
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# advertisement-interval 300
```

Configuring BGP Reconnect Interval

You can configure an interval after which a BGP session can reconnect.

BEFORE YOU BEGIN

Ensure that you have enabled the BGP feature.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **reconnect-interval** *interval*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 64496 switch(config-router)#	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.
Step 3	reconnect-interval <i>interval</i> Example: switch(config-router)# reconnect-interval 20 switch(config-router)#	Configures the interval after which a dropped BGP connection can automatically reconnect. The <i>interval</i> range is from 1 second to 60 seconds. The default value of the interval is 30 seconds.
Step 4	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure the BGP reconnect interval:

```
switch# configure terminal
switch(config)# router bgp 64496
switch(config-router)# reconnect-interval 20
switch(config-router)# copy running-config startup-config
```

Configuring Dynamic AS Numbers for Prefix Peers

You can configure multiple BGP peers within a BGP process. You can limit BGP session establishment to a single AS number or multiple AS numbers in a route map.

BGP sessions configured through dynamic AS numbers for prefix peers ignore the **ebgp-multihop** command and the **disable-connected-check** command.

You can change the list of AS numbers in the route map, but you must use the **no neighbor** command to change the route-map name. Changes to the AS numbers in the configured route map affect only new sessions.

BEFORE YOU BEGIN

Ensure that you have enabled the BGP feature.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor** *prefix* **remote-as** **route-map** *map-name*
4. (Optional) **show bgp ipv4 unicast neighbors**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 64496 switch(config-router)#	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.
Step 3	neighbor <i>prefix</i> remote-as route-map <i>map-name</i> Example: switch(config-router)# neighbor 192.0.2.0/8 remote-as routemap BGPPeers switch(config-router-neighbor)#	Configures the IPv4 prefix and a route map for the list of accepted AS numbers for the remote BGP peers. The <i>prefix</i> format for IPv4 is x.x.x.x/length. The length range is from 1 to 32. The <i>map-name</i> can be any case-sensitive, alphanumeric string up to 63 characters.

	Command	Purpose
Step 4	show bgp {ipv4 unicast neighbors} Example: switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors	(Optional) Displays information about BGP peers.
Step 5	copy running-config startup-config Example: switch(config-router-neighbor-af) copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure dynamic AS numbers for a prefix peer:

```

switch# configure terminal
switch(config)# route-map BGPPeers
switch(config-route-map)# match as-number 64496, 64501-64510
switch(config-route-map)# match as-number as-path-list List1, List2
switch(config-route-map)# exit
switch(config)# router bgp 64496
switch(config-router)# neighbor 192.0.2.0/8 remote-as route-map BGPPeers
switch(config-router-neighbor)# description Peer Router B
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# copy running-config startup-config

```

Clearing BGP Information

To clear BGP information, use the following commands:

Command	Purpose
clear bgp all { <i>neighbor</i> * <i>as-number</i> peer-template <i>name</i> <i>prefix</i> } [vrf <i>vrf-name</i>]	Clears one or more neighbors from all address families. * clears all neighbors in all address families. The arguments are as follows: <ul style="list-style-type: none"> <i>neighbor</i>—IPv4 address of a neighbor. <i>as-number</i>—Autonomous system number. The AS number can be a 16-bit integer or a 32-bit integer in the form of higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. <i>name</i>—Peer template name. The name can be any case-sensitive, alphanumeric string up to 64 characters. <i>prefix</i>—IPv4 prefix. All neighbors within that prefix are cleared. <i>vrf-name</i>—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.
clear bgp all dampening [vrf <i>vrf-name</i>]	Clears route flap dampening networks in all address families. The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 64 characters.
clear bgp all flap-statistics [vrf <i>vrf-name</i>]	Clears route flap statistics in all address families. The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 64 characters.
clear bgp ipv4unicast dampening [vrf <i>vrf-name</i>]	Clears route flap dampening networks in the selected address family. The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 64 characters.
clear bgp ipv4 unicast flap-statistics [vrf <i>vrf-name</i>]	Clears route flap statistics in the selected address family. The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 64 characters.

Command	Purpose
clear bgp { ipv4 ipv6 } unicast { <i>neighbor</i> * <i>as-number</i> peer-template <i>name</i> <i>prefix</i> } [vrf <i>vrf-name</i>]	<p>Clears one or more neighbors from the selected address family. * clears all neighbors in the address family. The arguments are as follows:</p> <ul style="list-style-type: none"> <i>neighbor</i>—IPv4 address of a neighbor. <i>as-number</i>— Autonomous system number. The AS number can be a 16-bit integer or a 32-bit integer in the form of higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. <i>name</i>—Peer template name. The name can be any case-sensitive, alphanumeric string up to 64 characters. <i>prefix</i>—IPv4 prefix. All neighbors within that prefix are cleared. <i>vrf-name</i>—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.
clear ip bgp { ip unicast { <i>neighbor</i> * <i>as-number</i> peer-template <i>name</i> <i>prefix</i> } [vrf <i>vrf-name</i>]	<p>Clears one or more neighbors. * clears all neighbors in the address family. The arguments are as follows:</p> <ul style="list-style-type: none"> <i>neighbor</i>—IPv4 address of a neighbor. <i>as-number</i>— Autonomous system number. The AS number can be a 16-bit integer or a 32-bit integer in the form of higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. <i>name</i>—Peer template name. The name can be any case-sensitive, alphanumeric string up to 64 characters. <i>prefix</i>—IPv4 prefix. All neighbors within that prefix are cleared. <i>vrf-name</i>—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.

Command	Purpose
clear ip bgp dampening [<i>ip-neighbor</i> <i>ip-prefix</i>] [vrf <i>vrf-name</i>]	<p>Clears route flap dampening in one or more networks. The arguments are as follows:</p> <ul style="list-style-type: none"> <i>ip-neighbor</i>—IPv4 or IPv6 address of a neighbor. <i>ip-prefix</i>—IPv4 or IPv6. All neighbors within that prefix are cleared. <i>vrf-name</i>—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.
clear ip bgp flap-statistics [<i>ip-neighbor</i> <i>ip-prefix</i>] [vrf <i>vrf-name</i>]	<p>Clears route flap statistics in one or more networks. The arguments are as follows:</p> <ul style="list-style-type: none"> <i>ip-neighbor</i>—IPv4 or IPv6 address of a neighbor. <i>ip-prefix</i>—IPv4 or IPv6. All neighbors within that prefix are cleared. <i>vrf-name</i>—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.
clear ip mbgp { ip unicast { <i>neighbor</i> * <i>as-number</i> peer-template <i>name</i> <i>prefix</i> } [vrf <i>vrf-name</i>]	<p>Clears one or more neighbors. * clears all neighbors in the address family. The arguments are as follows:</p> <ul style="list-style-type: none"> <i>neighbor</i>—IPv4 address of a neighbor. <i>as-number</i>— Autonomous system number. The AS number can be a 16-bit integer or a 32-bit integer in the form of higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. <i>name</i>—Peer template name. The name can be any case-sensitive, alphanumeric string up to 64 characters. <i>prefix</i>—IPv4 prefix. All neighbors within that prefix are cleared. <i>vrf-name</i>—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.

Command	Purpose
clear ip mbgp dampening [<i>ip-neighbor</i> <i>ip-prefix</i>] [vrf <i>vrf-name</i>]	Clears route flap dampening in one or more networks. The arguments are as follows: <ul style="list-style-type: none"> <i>ip-neighbor</i>—IPv4 or IPv6 address of a neighbor. <i>ip-prefix</i>—IPv4 or IPv6. All neighbors within that prefix are cleared. <i>vrf-name</i>—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.
clear ip mbgp flap-statistics [<i>ip-neighbor</i> <i>ip-prefix</i>] [vrf <i>vrf-name</i>]	Clears route flap statistics one or more networks. The arguments are as follows: <ul style="list-style-type: none"> <i>ip-neighbor</i>—IPv4 or IPv6 address of a neighbor. <i>ip-prefix</i>—IPv4 or IPv6. All neighbors within that prefix are cleared. <i>vrf-name</i>—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.

Verifying the Basic BGP Configuration

To display the BGP configuration information, perform the following tasks:

Command	Purpose
show bgp all [summary] [vrf <i>vrf-name</i>]	Displays the BGP information for all address families.
show bgp convergence [vrf <i>vrf-name</i>]	Displays the BGP information for all address families.
show bgp { ipv4 unicast [<i>ip-address</i>] community { regex <i>expression</i> community } [no-advertise] [no-export] [no-export-subconfed] } [vrf <i>vrf-name</i>]	Displays the BGP routes that match a BGP community.
show bgp [vrf <i>vrf-name</i>] { ip ipv6 } unicast [<i>ip-address</i>] community-list <i>list-name</i> [vrf <i>vrf-name</i>]	Displays the BGP routes that match a BGP community list.
show bgp ip unicast [<i>ip-address</i>] extcommunity { regex <i>expression</i> generic [non-transitive transitive] <i>aa4:nn</i> [exact-match] } [vrf <i>vrf-name</i>]	Displays the BGP routes that match a BGP extended community.

Command	Purpose
show bgp ip unicast [<i>ip-address</i>] extcommunity-list <i>list-name</i> [exact-match] [vrf <i>vrf-name</i>]	Displays the BGP routes that match a BGP extended community list.
show bgp ip unicast [<i>ip-address</i>] { dampening dampened-paths [regex <i>expression</i>]} [vrf <i>vrf-name</i>]	Displays the information for BGP route dampening. Use the clear bgp dampening command to clear the route flap dampening information.
show bgp ip unicast [<i>ip-address</i>] history-paths [regex <i>expression</i>] [vrf <i>vrf-name</i>]	Displays the BGP route history paths.
show bgp ip unicast [<i>ip-address</i>] filter-list <i>list-name</i> [vrf <i>vrf-name</i>]	Displays the information for the BGP filter list.
show bgp ip unicast [<i>ip-address</i>] neighbors [<i>ip-address</i>] [vrf <i>vrf-name</i>]	Displays the information for BGP peers. Use the clear bgp neighbors command to clear these neighbors.
show bgp ip unicast [<i>ip-address</i>] { nexthop nexthop-database } [vrf <i>vrf-name</i>]	Displays the information for the BGP route next-hop.
show bgp paths	Displays the BGP path information.
show bgp ip unicast [<i>ip-address</i>] policy name [vrf <i>vrf-name</i>]	Displays the BGP policy information. Use the clear bgp policy command to clear the policy information.
show bgp ip unicast [<i>ip-address</i>] prefix-list <i>list-name</i> [vrf <i>vrf-name</i>]	Displays the BGP routes that match the prefix list.
show bgp ip unicast [<i>ip-address</i>] received-paths [vrf <i>vrf-name</i>]	Displays the BGP paths stored for soft reconfiguration.
show bgp ip unicast [<i>ip-address</i>] regex <i>expression</i> [vrf <i>vrf-name</i>]	Displays the BGP routes that match the AS_path regular expression.
show bgp ip unicast [<i>ip-address</i>] route-map <i>map-name</i> [vrf <i>vrf-name</i>]	Displays the BGP routes that match the route map.
show bgp peer-policy name [vrf <i>vrf-name</i>]	Displays the information about BGP peer policies.
show bgp peer-session name [vrf <i>vrf-name</i>]	Displays the information about BGP peer sessions.
show bgp peer-template name [vrf <i>vrf-name</i>]	Displays the information about BGP peer templates. Use the clear bgp peer-template command to clear all neighbors in a peer template.
show bgp process	Displays the BGP process information.
show ip bgp options	Displays the BGP status and configuration information. This command has multiple options. See the <i>Cisco Nexus 3000 Series Command Reference</i> for more information.

Command	Purpose
<code>show ip mbgp options</code>	Displays the BGP status and configuration information. This command has multiple options. See the <i>Cisco Nexus 3000 Series Command Reference</i> for more information.
<code>show running-configuration bgp</code>	Displays the current running BGP configuration.

Displaying BGP Statistics

To display BGP statistics, use the following commands:

Command	Purpose
<code>show bgp ip unicast [ip-address] flap-statistics [vrf vrf-name]</code>	Displays the BGP route flap statistics. Use the clear bgp flap-statistics command to clear these statistics.
<code>show bgp sessions [vrf vrf-name]</code>	Displays the BGP sessions for all peers. Use the clear bgp sessions command to clear these statistics.
<code>show bgp sessions [vrf vrf-name]</code>	Displays the BGP sessions for all peers. Use the clear bgp sessions command to clear these statistics.
<code>show bgp statistics</code>	Displays the BGP statistics.

Configuration Examples for Basic BGP

This example shows a basic BGP configuration:

```
feature bgp
router bgp 64496
 neighbor 192.0.2.1 remote-as 64496
  address-family ipv4 unicast
    next-hop-self
```

Related Topics

The following topics relate to BGP:

- [Chapter 16, “Configuring Route Policy Manager”](#)

Where to Go Next

See [Chapter 8, “Configuring Advanced BGP”](#) for details on the following features:

- Peer templates
- Route redistribution
- Route maps

Additional References

For additional information related to implementing BGP, see the following sections:

- [Related Documents, page 7-28](#)
- [MIBs, page 7-28](#)

Related Documents

Related Topic	Document Title
BGP CLI commands	<i>Cisco Nexus 3000 Series Command Reference</i>

MIBs

MIBs	MIBs Link
BGP4-MIB	To locate and download MIBs, go to the following URL:
CISCO-BGP4-MIB	http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml



Configuring Advanced BGP

This chapter describes how to configure advanced features of the Border Gateway Protocol (BGP) on the Cisco NX-OS switch.

This chapter includes the following sections:

- [Information About Advanced BGP, page 8-1](#)
- [Licensing Requirements for Advanced BGP, page 8-10](#)
- [Prerequisites for Advanced BGP, page 8-10](#)
- [Guidelines and Limitations for Advanced BGP, page 8-11](#)
- [Default Settings, page 8-12](#)
- [Configuring Advanced BGP, page 8-12](#)
- [Verifying the Advanced BGP Configuration, page 8-48](#)
- [Displaying BGP Statistics, page 8-49](#)
- [Related Topics, page 8-49](#)
- [Additional References, page 8-50](#)

Information About Advanced BGP

BGP is an interdomain routing protocol that provides loop-free routing between organizations or autonomous systems. Cisco NX-OS supports BGP version 4. BGP version 4 includes multiprotocol extensions that allow BGP to carry routing information and multiple Layer 3 protocol address families. BGP uses TCP as a reliable transport protocol to create TCP sessions with other BGP-enabled switches called BGP peers. When connecting to an external organization, the router creates external BGP (eBGP) peering sessions. BGP peers within the same organization exchange routing information through internal BGP (iBGP) peering sessions.

This section includes the following topics:

- [Peer Templates, page 8-2](#)
- [Authentication, page 8-2](#)
- [Route Policies and Resetting BGP Sessions, page 8-3](#)
- [eBGP, page 8-3](#)
- [eBGP Next-Hop Unchanged, page 8-4](#)
- [iBGP, page 8-4](#)

- [Capabilities Negotiation, page 8-6](#)
- [Route Dampening, page 8-6](#)
- [Load Sharing and Multipath, page 8-7](#)
- [Route Aggregation, page 8-7](#)
- [BGP Conditional Advertisement, page 8-7](#)
- [BGP Next-Hop Address Tracking, page 8-8](#)
- [Site of Origin, page 8-8](#)
- [BFD, page 8-9](#)
- [Tuning BGP, page 8-9](#)
- [Multiprotocol BGP, page 8-10](#)
- [Licensing Requirements for Advanced BGP, page 8-10](#)

Peer Templates

BGP peer templates allow you to create blocks of common configurations that you can reuse across similar BGP peers. Each block allows you to define a set of attributes that a peer then inherits. You can choose to override some of the inherited attributes as well, making it a very flexible scheme for simplifying the repetitive nature of BGP configurations.

Cisco NX-OS implements three types of peer templates:

- The *peer-session* template defines BGP peer session attributes, such as the transport details, remote autonomous system number of the peer, and session timers. A peer-session template can also inherit attributes from another peer-session template (with locally defined attributes that override the attributes from an inherited peer-session).
- A *peer-policy* template defines the address-family dependent policy aspects for a peer including the inbound and outbound policy, filter-lists, and prefix-lists. A peer-policy template can inherit from a set of peer-policy templates. Cisco NX-OS evaluates these peer-policy templates in the order specified by the preference value in the inherit configuration. The lowest number is preferred over higher numbers.
- The *peer* template can inherit the peer-session and peer-policy templates to allow for simplified peer definitions. It is not mandatory to use a peer template but it can simplify the BGP configuration by providing reusable blocks of configuration.

Authentication

You can configure authentication for a BGP neighbor session. This authentication method adds an MD5 authentication digest to each TCP segment sent to the neighbor to protect BGP against unauthorized messages and TCP security attacks.



Note

The MD5 password must be identical between BGP peers.

Route Policies and Resetting BGP Sessions

You can associate a route policy to a BGP peer. Route policies use route maps to control or modify the routes that BGP recognizes. You can configure a route policy for inbound or outbound route updates. The route policies can match on different criteria, such as a prefix or AS_path attribute, and selectively accept or deny the routes. Route policies can also modify the path attributes.

When you change a route policy applied to a BGP peer, you must reset the BGP sessions for that peer. Cisco NX-OS supports the following three mechanisms to reset BGP peering sessions:

- **Hard reset**—A hard reset tears down the specified peering sessions, including the TCP connection, and deletes routes coming from the specified peer. This option interrupts packet flow through the BGP network. Hard reset is disabled by default.
- **Soft reconfiguration inbound**—A soft reconfiguration inbound triggers routing updates for the specified peer without resetting the session. You can use this option if you change an inbound route policy. Soft reconfiguration inbound saves a copy of all routes received from the peer before processing the routes through the inbound route policy. If you change the inbound route policy, Cisco NX-OS passes these stored routes through the modified inbound route policy to update the route table without tearing down existing peering sessions. Soft reconfiguration inbound can use significant memory resources to store the unfiltered BGP routes. Soft reconfiguration inbound is disabled by default.
- **Route Refresh**—A route refresh updates the inbound routing tables dynamically by sending route refresh requests to supporting peers when you change an inbound route policy. The remote BGP peer responds with a new copy of its routes that the local BGP speaker processes with the modified route policy. Cisco NX-OS automatically sends an outbound route refresh of prefixes to the peer.

BGP peers advertise the route refresh capability as part of the BGP capability negotiation when establishing the BGP peer session. Route refresh is the preferred option and enabled by default.

**Note**

BGP also uses route maps for route redistribution, route aggregation, route dampening, and other features. See [Chapter 16, “Configuring Route Policy Manager,”](#) for more information on route maps.

eBGP

External BGP (eBGP) allows you to connect BGP peers from different autonomous systems to exchange routing updates. Connecting to external networks enables traffic from your network to be forwarded to other networks and across the Internet.

You should use loopback interfaces for establishing eBGP peering sessions because loopback interfaces are less susceptible to interface flapping. An interface *flap* occurs when the interface is administratively brought up or down because of a failure or maintenance issue. See the [“Configuring eBGP” section on page 8-25](#) for information on multihop, fast external failovers, and limiting the size of the AS-path attribute.

eBGP Next-Hop Unchanged

In an external BGP (eBGP) session, by default, the device changes the next-hop attribute of a BGP route (to its own address) when the device sends out a route. If the eBGP Next-Hop Unchanged feature is configured, BGP sends routes to an eBGP multihop peer without modifying the next-hop attribute. The next-hop attribute is unchanged. The BGP Next-hop Unchanged feature provides flexibility when designing and migrating networks. It can be used only between eBGP peers configured as multihop.

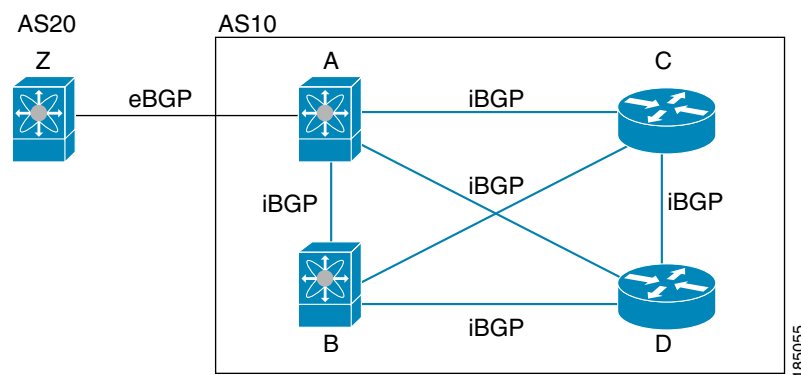
For example, consider a network with eBGP connection between Devices A, B, and C. Suppose Device A announces 100 prefixes to Device B. Device B is configured with an outbound route map to Device C and the match ip prefix list and set ip next-hop unchanged are configured on the route map. Device B propagates the unchanged next-hop address only for the routes that match the prefix list. For the other prefixes, it puts itself as the next-hop address.

iBGP

Internal BGP (iBGP) allows you to connect BGP peers within the same autonomous system. You can use iBGP for multihomed BGP networks (networks that have more than one connection to the same external autonomous system).

Figure 8-1 shows an iBGP network within a larger BGP network.

Figure 8-1 iBGP Network



iBGP networks are fully meshed. Each iBGP peer has a direct connection to all other iBGP peers to prevent network loops.



Note

You should configure a separate interior gateway protocol in the iBGP network.

This section includes the following topics:

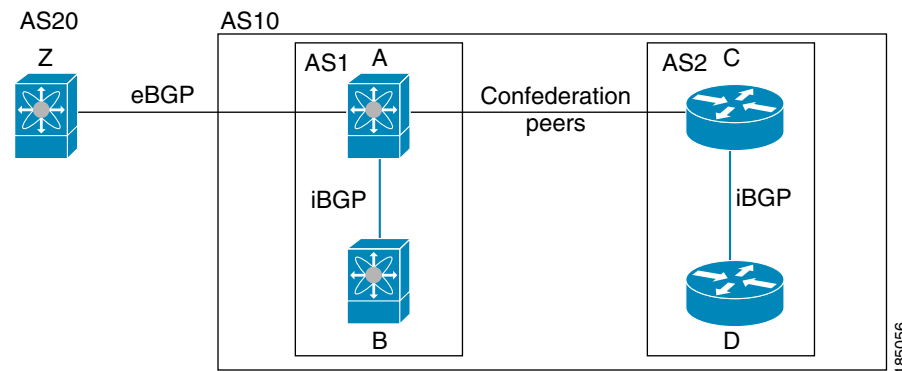
- [AS Confederations, page 8-5](#)
- [Route Reflector, page 8-5](#)

AS Confederations

A fully meshed iBGP network becomes complex as the number of iBGP peers grows. You can reduce the iBGP mesh by dividing the autonomous system into multiple subautonomous systems and grouping them into a single confederation. A confederation is a group of iBGP peers that use the same autonomous system number to communicate to external networks. Each subautonomous system is fully meshed within itself and has a few connections to other subautonomous systems in the same confederation.

Figure 8-2 shows the BGP network from Figure 8-1, split into two subautonomous systems and one confederation.

Figure 8-2 AS Confederation



In this example, AS10 is split into two subautonomous systems, AS1 and AS2. Each subautonomous system is fully meshed, but there is only one link between the subautonomous systems. By using AS confederations, you can reduce the number of links compared to the fully meshed autonomous system in Figure 8-1.

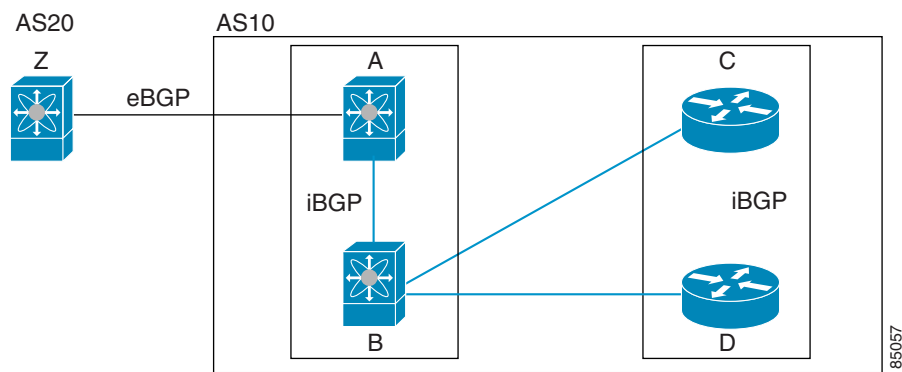
Route Reflector

You can alternately reduce the iBGP mesh by using a route reflector configuration. Route reflectors pass learned routes to neighbors so that all iBGP peers do not need to be fully meshed.

Figure 8-1 shows a simple iBGP configuration with four meshed iBGP speakers (router A, B, C, and D). Without route reflectors, when router A receives a route from an external neighbor, it advertises the route to all three iBGP neighbors.

When you configure an iBGP peer to be a route reflector, it becomes responsible for passing iBGP learned routes to a set of iBGP neighbors.

In Figure 8-3, router B is the route reflector. When the route reflector receives routes advertised from router A, it advertises (reflects) the routes to routers C and D. Router A no longer has to advertise to both routers C and D.

Figure 8-3 Route Reflector

The route reflector and its client peers form a cluster. You do not have to configure all iBGP peers to act as client peers of the route reflector. You must configure any nonclient peer as fully meshed to guarantee that complete BGP updates reach all peers.

Capabilities Negotiation

A BGP speaker can learn about BGP extensions supported by a peer by using the capabilities negotiation feature. Capabilities negotiation allows BGP to use only the set of features supported by both BGP peers on a link.

If a BGP peer does not support capabilities negotiation, Cisco NX-OS will attempt a new session to the peer without capabilities negotiation if you have configured the address family as IPv4.

Route Dampening

Route dampening is a BGP feature that minimizes the propagation of flapping routes across an internetwork. A route flaps when it alternates between the available and unavailable states in rapid succession.

For example, consider a network with three BGP autonomous systems: AS1, AS2, and AS3. Suppose that a route in AS1 flaps (it becomes unavailable). Without route dampening, AS1 sends a withdraw message to AS2. AS2 propagates the withdrawal message to AS3. When the flapping route reappears, AS1 sends an advertisement message to AS2, which sends the advertisement to AS3. If the route repeatedly becomes unavailable, and then available, AS1 sends many withdrawal and advertisement messages that propagate through the other autonomous systems.

Route dampening can minimize flapping. Suppose that the route flaps. AS2 (in which route dampening is enabled) assigns the route a penalty of 1000. AS2 continues to advertise the status of the route to neighbors. Each time that the route flaps, AS2 adds to the penalty value. When the route flaps so often that the penalty exceeds a configurable suppression limit, AS2 stops advertising the route, regardless of how many times that it flaps. The route is now dampened.

The penalty placed on the route decays until the reuse limit is reached. At that time, AS2 advertises the route again. When the reuse limit is at 50 percent, AS2 removes the dampening information for the route.



Note

The router does not apply a penalty to a resetting BGP peer when route dampening is enabled, even though the peer reset withdraws the route.

Load Sharing and Multipath

BGP can install multiple equal-cost eBGP or iBGP paths into the routing table to reach the same destination prefix. Traffic to the destination prefix is then shared across all the installed paths.

The BGP best-path algorithm considers the paths as equal-cost paths if the following attributes are identical:

- Weight
- Local preference
- AS_path
- Origin code
- Multi-exit discriminator (MED)
- IGP cost to the BGP next-hop

BGP selects only one of these multiple paths as the best path and advertises the path to the BGP peers.

**Note**

Paths received from different AS confederations are considered as equal-cost paths if the external AS_path values and the other attributes are identical.

**Note**

When you configure a route reflector for iBGP multipath, and the route reflector advertises the selected best path to its peers, the next-hop for the path is not modified.

Route Aggregation

You can configure aggregate addresses. Route aggregation simplifies route tables by replacing a number of more specific addresses with an address that represents all the specific addresses. For example, you can replace these three more specific addresses, 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24 with one aggregate address, 10.1.0.0/16.

Aggregate prefixes are present in the BGP route table so that fewer routes are advertised.

**Note**

Cisco NX-OS does not support automatic route aggregation.

Route aggregation can lead to forwarding loops. To avoid this problem, when BGP generates an advertisement for an aggregate address, it automatically installs a summary discard route for that aggregate address in the local routing table. BGP sets the administrative distance of the summary discard to 220 and sets the route type to discard. BGP does not use discard routes for next-hop resolution.

BGP Conditional Advertisement

BGP conditional advertisement allows you to configure BGP to advertise or withdraw a route based on whether or not a prefix exists in the BGP table. This feature is useful, for example, in multihomed networks, in which you want BGP to advertise some prefixes to one of the providers only if information from the other provider is not present.

Consider an example network with three BGP autonomous systems: AS1, AS2, and AS3, where AS1 and AS3 connect to the Internet and to AS2. Without conditional advertisement, AS2 propagates all routes to both AS1 and AS3. With conditional advertisement, you can configure AS2 to advertise certain routes to AS3 only if routes from AS1 do not exist (if for example, the link to AS1 fails).

BGP conditional advertisement adds an exist or not-exist test to each route that matches the configured route map. See the [“Suppressing BGP Routes” section on page 8-35](#) for more information.

BGP Next-Hop Address Tracking

BGP monitors the next-hop address of installed routes to verify next-hop reachability and to select, install, and validate the BGP best path. BGP next-hop address tracking speeds up this next-hop reachability test by triggering the verification process when routes change in the RIB that may affect BGP next-hop reachability.

BGP receives notifications from the RIB when next-hop information changes (event-driven notifications). BGP is notified when any of the following events occurs:

- Next hop becomes unreachable.
- Next hop becomes reachable.
- Fully recursed Interior Gateway Protocol (IGP) metric to the next-hop changes.
- First hop IP address or first hop interface changes.
- Next hop becomes connected.
- Next hop becomes unconnected.
- Next hop becomes a local address.
- Next hop becomes a nonlocal address.



Note

Reachability and recursed metric events trigger a best-path recalculation.

Event notifications from the RIB are classified as critical and noncritical. Notifications for critical and noncritical events are sent in separate batches. However, a noncritical event is sent with the critical events if the noncritical event is pending and there is a request to read the critical events.

- Critical events are related to the reachability (reachable and unreachable), connectivity (connected and unconnected), and locality (local and nonlocal) of the next-hops. Notifications for these events are not delayed.
- Noncritical events include only the IGP metric changes.

See the [“Configuring BGP Next-Hop Address Tracking” section on page 8-23](#) for more information.

Site of Origin

The site of origin prevents routing loops when you have a multihomed VPN site. Routes learned from the same site are tagged with the same site-of-origin value that is configured at the PE on all the PE-CE links to the same site. Routes with a particular site-of-origin value are never readvertised back to a CE with the same site-of-origin value configured at the PE-CE link. This process prevents a CE router from relearning routes that originated from the same site. BGP and EIGRP use site of origin to prevent loops.

You can override the autonomous system number (ASN) of a site with the ASN of the provider. This feature is often used with the site of origin to identify the site where a route originated and prevent routing loops between routers within a VPN.

Route Redistribution

You can configure BGP to redistribute static routes or routes from other protocols. You configure a route policy with the **redistribution to control which routes are passed into BGP**. A route policy allows you to filter routes based on attributes such as the destination, origination protocol, route type, route tag, and so on. See [Chapter 16, “Configuring Route Policy Manager,”](#) for more information.

BFD

This feature supports bidirectional forwarding detection (BFD). BFD is a detection protocol designed to provide fast forwarding-path failure detection times. 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.

BFD for BGP is supported on eBGP single-hop peers and iBGP single-hop peers. For iBGP single-hop peers using BFD, you must configure the update-source option in neighbor configuration mode. BFD is not supported on other iBGP peers or for multihop eBGP peers.

BFD is supported for the following types of interfaces:

- Layer 3 physical and subinterface
- Layer 3 port channel and subinterface
- Switch virtual interface (SVI)

BFD for BGP does not support authentication or per-link BFD sessions on a port channel.

See [Chapter 18, “Configuring Bidirectional Forwarding Detection”](#) for more information.

Tuning BGP

You can modify the default behavior of BGP through BGP timers and by adjusting the best-path algorithm.

This section includes the following topics:

- [BGP Timers, page 8-9](#)
- [Tuning the Best-Path Algorithm, page 8-10](#)

BGP Timers

BGP uses different types of timers for neighbor session and global protocol events. Each established session has a minimum of two timers for sending periodic keepalive messages and for timing out sessions when peer keepalives do not arrive within the expected time. In addition, there are other timers for handling specific features. Typically, you configure these timers in seconds. The timers include a random adjustment so that the same timers on different BGP peers trigger at different times.

Tuning the Best-Path Algorithm

You can modify the default behavior of the best-path algorithm through optional configuration parameters, including changing how the algorithm handles the multi-exit discriminator attribute and the router ID.

Multiprotocol BGP

BGP on Cisco NX-OS supports multiple address families. Multiprotocol BGP (MP-BGP) carries different sets of routes depending on the address family. For example, BGP can carry one set of routes for IPv4 unicast routing and IPv6 unicast routing.

Use the router address-family and neighbor address-family configuration modes to support multiprotocol BGP configurations. MP-BGP maintains separate RIBs for each configured address family, such as a unicast RIB for BGP.

RFC 5549

Beginning with Cisco NX-OS Release 6.0(2)U4(1), BGP supports RFC 5549, which allows an IPv4 prefix to be carried over an IPv6 next hop. Because BGP is running on every hop, and all routers are capable of forwarding IPv4 and IPv6 traffic, there is no need to support IPv6 tunnels between any routers. BGP installs IPv4 over an IPv6 route to the Unicast Route Information Base (URIB).

Licensing Requirements for Advanced BGP

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	BGP requires an LAN Enterprise Services license. For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the <i>Cisco NX-OS Licensing Guide</i> .
	Note Make sure the LAN Base Services license is installed on the switch to enable Layer 3 interfaces.

Prerequisites for Advanced BGP

BGP has the following prerequisites:

- You must enable the BGP feature (see the [“Enabling the BGP Feature”](#) section on page 7-11).
- You should have a valid router ID configured on the system.
- You must have an AS number, either assigned by a Regional Internet Registry (RIR) or locally administered.
- You must have reachability (such as an interior gateway protocol (IGP), a static route, or a direct connection) to the peer that you are trying to make a neighbor relationship with.
- You must explicitly configure an address family under a neighbor for the BGP session establishment.

Guidelines and Limitations for Advanced BGP

BGP has the following configuration guidelines and limitations:

- The dynamic AS number prefix peer configuration overrides the individual AS number configuration inherited from a BGP template.
- If you configure a dynamic AS number for prefix peers in an AS confederation, BGP establishes sessions with only the AS numbers in the local confederation.
- BGP sessions created through a dynamic AS number prefix peer ignore any configured eBGP multihop time-to-live (TTL) value or a disabled check for directly connected peers.
- Configure a router ID for BGP to avoid automatic router ID changes and session flaps.
- Use the maximum-prefix configuration option per peer to restrict the number of routes received and system resources used.
- Configure the update-source to establish a session with eBGP multihop sessions.
- Specify a BGP route map if you configure redistribution.
- Configure the BGP router ID within a VRF.
- If you decrease the keepalive and hold timer values, the network might experience session flaps.
- The following guidelines and limitations apply to the **remove-private-as** command
 - If the local AS number of the device is a private AS number, you cannot use the **remove-private-as** configuration command for any other neighbor on the same device. As a workaround, you can use the **local-as** command on each neighbor with a public local AS number.
 - If the real AS number of the device is a private AS number and the **remove-private-as all** command is configured for a neighbor with a public local-as number, use **local-as number [no-prepend [replace-as]]** command to ensure that the real private AS number is not appended to the AS path.
 - If the real AS number of the device is a public AS number and the **remove-private-as all** command is configured for a neighbor, you cannot configure a private local-as number for the same neighbor. As a workaround, you must remove the existing configuration to proceed further.
 - The **remove-private-as all** command removes private AS numbers from the AS path even if the path contains both public and private AS numbers.
 - The **remove-private-as** command removes private AS numbers even if the AS path contains only private AS numbers. There is no likelihood of a 0-length AS path because this command can be applied to eBGP peers only, in which case the AS number of the local device is appended to the AS path.
 - The **remove-private-as** command removes private AS numbers even if the private AS numbers appear before the confederation segments in the AS path.
 - When you remove private AS numbers from the AS path, the path length of the prefixes that are sent out will decrease. Because the AS path length is a key element of BGP best-path selection, it might be necessary to retain the path length. The **replace-as** keyword ensures that the path length is retained by replacing all removed AS numbers with the local router's AS number.

Default Settings

Table 8-1 lists the default settings for BGP parameters.

Table 8-1 *Default BGP Parameters*

Parameters	Default
BGP feature	Disabled
keep alive interval	60 seconds
hold timer	180 seconds

Configuring Advanced BGP

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

- [Enabling IP Forward on an Interface, page 8-13](#)
- [Enabling IP Forward on an Interface, page 8-13](#)
- [Configuring BGP Peer-Policy Templates, page 8-16](#)
- [Configuring BGP Peer Templates, page 8-18](#)
- [Configuring Prefix Peering, page 8-21](#)
- [Configuring BGP Authentication, page 8-22](#)
- [Resetting a BGP Session, page 8-22](#)
- [Modifying the Next-Hop Address, page 8-23](#)
- [Configuring BGP Next-Hop Address Tracking, page 8-23](#)
- [Configuring Next-Hop Filtering, page 8-24](#)
- [Shrinking Next-Hop Groups When A Session Goes Down, page 8-24](#)
- [Configuring eBGP, page 8-25](#)
- [Configuring AS Confederations, page 8-29](#)
- [Configuring Route Reflector, page 8-31](#)
- [Configuring Route Dampening, page 8-33](#)
- [Configuring Route Deletion Delay, page 8-33](#)
- [Configuring Maximum Prefixes, page 8-34](#)
- [Configuring Dynamic Capability, page 8-34](#)
- [Configuring Aggregate Addresses, page 8-35](#)
- [Suppressing BGP Routes, page 8-35](#)
- [Configuring Route Redistribution, page 8-38](#)
- [Disabling BGP Dampening with Redistribution, page 8-39](#)
- [Configuring Multiprotocol BGP, page 8-40](#)
- [Configuring BGP Extended Community Site of Origin, page 8-42](#)
- [Configuring Virtualization, page 8-46](#)

**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 IP Forward on an Interface

To use RFC 5549, you must configure at least one IPv4 address. If you do not want to configure an IPv4 address, you must enable the `ip forward` feature to use RFC 5549.

SUMMARY STEPS

1. **`configure terminal`**
2. **`interface type slot/port`**
3. **`ip forward`**

DETAILED STEPS

	Command	Purpose
Step 1	<code>configure terminal</code> Example: switch# <code>configure terminal</code> switch(config)#	Enters global configuration mode.
Step 2	<code>interface type slot/port</code> Example: switch(config)# <code>interface ethernet 1/2</code> switch(config-if)#	Enters interface configuration mode.
Step 3	<code>ip forward</code> Example: switch(config-if)# <code>ip forward</code> switch(config-if)#	Allows IPv4 traffic on the interface even when there is no IP address configuration on that interface.

This example shows how to enable the **`ip forward`** feature on an interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ip forward
```

Configuring BGP Session Templates

You can use BGP session templates to simplify BGP configuration for multiple BGP peers with similar configuration needs. BGP templates allow you to reuse common configuration blocks. You configure BGP templates first, and then apply these templates to BGP peers.

With BGP session templates, you can configure session attributes such as inheritance, passwords, timers, and security.

A peer-session template can inherit from one other peer-session template. You can configure the second template to inherit from a third template. The first template also inherits this third template. This indirect inheritance can continue for up to seven peer-session templates.

Any attributes configured for the neighbor take priority over any attributes inherited by that neighbor from a BGP template.

BEFORE YOU BEGIN

Ensure that you have enabled the BGP feature (see the [“Enabling the BGP Feature”](#) section on page 7-11).



Note

When editing a template, you can use the **no** form of a command at either the peer or template level to explicitly override a setting in a template. You must use the **default** form of the command to reset that attribute to the default state.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **template peer-session** *template-name*
4. **password** *number password*
5. **timers** *keepalive hold*
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **inherit peer-session** *template-name*
9. (Optional) **description** *text*
10. (Optional) **show bgp peer-session** *template-name*
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 65536 switch(config-router)#	Enables BGP and assigns the autonomous system number to the local BGP speaker.

	Command	Purpose
Step 3	template peer-session <i>template-name</i> Example: switch(config-router)# template peer-session BaseSession switch(config-router-stmp)#	Enters peer-session template configuration mode.
Step 4	password <i>number password</i> Example: switch(config-router-stmp)# password 0 test	(Optional) Adds the cleartext password <i>test</i> to the neighbor. The password is stored and displayed in type 3 encrypted form (3DES).
Step 5	timers <i>keepalive hold</i> Example: switch(config-router-stmp)# timers 30 90	(Optional) Adds the BGP keepalive and holdtimer values to the peer-session template. The default keepalive interval is 60. The default hold time is 180.
Step 6	exit Example: switch(config-router-stmp)# exit switch(config-router)#	Exits peer-session template configuration mode.
Step 7	neighbor <i>ip-address remote-as as-number</i> Example: switch(config-router)# neighbor 192.168.1.2 remote-as 65536 switch(config-router-neighbor)#	Places the router in the neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 8	inherit peer-session <i>template-name</i> Example: switch(config-router-neighbor)# inherit peer-session BaseSession switch(config-router-neighbor)	Applies a peer-session template to the peer.
Step 9	description <i>text</i> Example: switch(config-router-neighbor)# description Peer Router A switch(config-router-neighbor)	(Optional) Adds a description for the neighbor.
Step 10	show bgp peer-session <i>template-name</i> Example: switch(config-router-neighbor)# show bgp peer-session BaseSession	(Optional) Displays the peer-policy template.
Step 11	copy running-config startup-config Example: switch(config-router-neighbor)# copy running-config startup-config	(Optional) Saves this configuration change.

Use the **show bgp neighbor** command to see the template applied. See the *Cisco Nexus 3000 Series Command Reference* for details on all commands available in the template.

This example shows how to configure a BGP peer-session template and apply it to a BGP peer:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer-session BaseSession
switch(config-router-stmp)# timers 30 90
switch(config-router-stmp)# exit
switch(config-router)# neighbor 192.168.1.2 remote-as 65536
switch(config-router-neighbor)# inherit peer-session BaseSession
switch(config-router-neighbor)# description Peer Router A
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor)# copy running-config startup-config
```

Configuring BGP Peer-Policy Templates

You can configure a peer-policy template to define attributes for a particular address family. You assign a preference to each peer-policy template and these templates are inherited in the order specified, for up to five peer-policy templates in a neighbor address family.

Cisco NX-OS evaluates multiple peer policies for an address family using the preference value. The lowest preference value is evaluated first. Any attributes configured for the neighbor take priority over any attributes inherited by that neighbor from a BGP template.

Peer-policy templates can configure address family-specific attributes such as AS-path filter lists, prefix lists, route reflection, and soft reconfiguration.

BEFORE YOU BEGIN

Ensure that you have enabled the BGP feature (see the [“Enabling the BGP Feature”](#) section on page 7-11).



Note

When editing a template, you can use the **no** form of a command at either the peer or template level to explicitly override a setting in a template. You must use the default form of the command to reset that attribute to the default state.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **template peer-policy** *template-name*
4. **advertise-active-only**
5. **maximum-prefix** *number*
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **address-family** **ipv4 unicast**
9. **inherit peer-policy** *template-name* *preference*
10. (Optional) **show bgp peer-policy** *template-name*
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 65536 switch(config-router)#	Enables BGP and assigns the autonomous system number to the local BGP speaker.
Step 3	template peer-policy <i>template-name</i> Example: switch(config-router)# template peer-policy BasePolicy switch(config-router-ptmp)#	Creates a peer-policy template.
Step 4	advertise-active-only Example: switch(config-router-ptmp)# advertise-active-only	(Optional) Advertises only active routes to the peer.
Step 5	maximum-prefix <i>number</i> Example: switch(config-router-ptmp)# maximum-prefix 20	(Optional) Sets the maximum number of prefixes allowed from this peer.
Step 6	exit Example: switch(config-router-ptmp)# exit switch(config-router)#	Exits peer-policy template configuration mode.
Step 7	neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: switch(config-router)# neighbor 192.168.1.2 remote-as 65536 switch(config-router-neighbor)#	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 8	address-family ipv4 unicast Example: switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#	Enters global address family configuration mode for the specified address family.
Step 9	inherit peer-policy <i>template-name</i> <i>preference</i> Example: switch(config-router-neighbor-af)# inherit peer-policy BasePolicy 1	Applies a peer-policy template to the peer address family configuration and assigns the preference value for this peer policy.

	Command	Purpose
Step 10	show bgp peer-policy <i>template-name</i> Example: switch(config-router-neighbor-af)# show bgp peer-policy BasePolicy	(Optional) Displays the peer-policy template.
Step 11	copy running-config startup-config Example: switch(config-router-neighbor-af)# copy running-config startup-config	(Optional) Saves this configuration change.

Use the **show bgp neighbor** command to see the template applied. See the *Cisco Nexus 3000 Series Command Reference* for details on all commands available in the template.

This example shows how to configure a BGP peer-session template and apply it to a BGP peer:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer-session BasePolicy
switch(config-router-ptmp)# maximum-prefix 20
switch(config-router-ptmp)# exit
switch(config-router)# neighbor 192.168.1.1 remote-as 65536
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# inherit peer-policy BasePolicy
switch(config-router-neighbor-af)# copy running-config startup-config
```

Configuring BGP Peer Templates

You can configure BGP peer templates to combine session and policy attributes in one reusable configuration block. Peer templates can also inherit peer-session or peer-policy templates. Any attributes configured for the neighbor take priority over any attributes inherited by that neighbor from a BGP template. You configure only one peer template for a neighbor, but that peer template can inherit peer-session and peer-policy templates.

Peer templates support session and address family attributes, such as eBGP multihop time-to-live, maximum prefix, next-hop self, and timers.

BEFORE YOU BEGIN

Ensure that you have enabled the BGP feature (see the [“Enabling the BGP Feature”](#) section on page 7-11).



Note

When editing a template, you can use the **no** form of a command at either the peer or template level to explicitly override a setting in a template. You must use the default form of the command to reset that attribute to the default state.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **template peer** *template-name*

4. (Optional) **inherit peer-session** *template-name*
5. (Optional) **address-family** {**ipv4** | **ipv6**} **unicast**
6. (Optional) **inherit peer** *template-name*
7. **exit**
8. (Optional) **timers** *keepalive hold*
9. **exit**
10. **neighbor** *ip-address remote-as as-number*
11. **inherit peer** *template-name*
12. (Optional) **timers** *keepalive hold*
13. (Optional) **show bgp peer-template** *template-name*
14. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 65536	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	template peer <i>template-name</i> Example: switch(config-router)# template peer BasePeer switch(config-router-neighbor)#	Enters peer template configuration mode.
Step 4	inherit peer-session <i>template-name</i> Example: switch(config-router-neighbor)# inherit peer-session BaseSession	(Optional) Inherits a peer-session template in the peer template.
Step 5	address-family ipv4 unicast Example: switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#	(Optional) Configures the global address family configuration mode for the specified address family.
Step 6	inherit peer <i>template-name</i> Example: switch(config-router-neighbor-af)# inherit peer BasePolicy	(Optional) Applies a peer template to the neighbor address family configuration.
Step 7	exit Example: switch(config-router-neighbor-af)# exit switch(config-router-neighbor)#	Exits BGP neighbor address family configuration mode.

	Command	Purpose
Step 8	timers <i>keepalive hold</i> Example: switch(config-router-neighbor)# timers 45 100	(Optional) Adds the BGP timer values to the peer. These values override the timer values in the peer-session template, BaseSession.
Step 9	exit Example: switch(config-router-neighbor)# exit switch(config-router)#	Exits BGP peer template configuration mode.
Step 10	neighbor <i>ip-address remote-as as-number</i> Example: switch(config-router)# neighbor 192.168.1.2 remote-as 65536 switch(config-router-neighbor)#	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 11	inherit peer <i>template-name</i> Example: switch(config-router-neighbor)# inherit peer BasePeer	Inherits the peer template.
Step 12	timers <i>keepalive hold</i> Example: switch(config-router-neighbor)# timers 60 120	(Optional) Adds the BGP timer values to this neighbor. These values override the timer values in the peer template and the peer-session template.
Step 13	show bgp peer-template <i>template-name</i> Example: switch(config-router-neighbor-af)# show bgp peer-template BasePeer	(Optional) Displays the peer template.
Step 14	copy running-config startup-config Example: switch(config-router-neighbor-af)# copy running-config startup-config	(Optional) Saves this configuration change.

Use the **show bgp neighbor** command to see the template applied. See the *Cisco Nexus 3000 Series Command Reference* for details on all commands available in the template.

This example shows how to configure a BGP peer template and apply it to a BGP peer:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer BasePeer
switch(config-router-neighbor)# inherit peer-session BaseSession
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# inherit peer-policy BasePolicy 1
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# exit
switch(config-router)# neighbor 192.168.1.2 remote-as 65536
switch(config-router-neighbor)# inherit peer BasePeer
switch(config-router-neighbor)# copy running-config startup-config
```


Configuring Prefix Peering

BGP supports the definition of a set of peers using a prefix for both IPv4 and IPv6. This feature allows you to not have to add each neighbor to the configuration.

When defining a prefix peering, you must specify the remote AS number with the prefix. BGP accepts any peer that connects from that prefix and autonomous system if the prefix peering does not exceed the configured maximum peers allowed.

When a BGP peer that is part of a prefix peering disconnects, Cisco NX-OS holds its peer structures for a defined prefix peer timeout value. An established peer can reset and reconnect without danger of being blocked because other peers have consumed all slots for that prefix peering.

To configure the BGP prefix peering timeout value, use the following command in router configuration mode:

Command	Purpose
timers prefix-peer-timeout <i>value</i> Example: switch(config-router)# timers prefix-peer-timeout 120	Configures the timeout value for prefix peering. The range is from 0 to 1200 seconds. The default value is 30.
timers prefix-peer-wait <i>interval</i> Example: switch(config-router)# timers prefix-peer-wait 50	(Optional) Configures the BGP prefix peering wait timer on a per-VRF basis or on the default VRF. You can use the timers prefix-peer-wait command to disable the peer prefix wait time so that there is no delay before BGP prefixes are inserted into the routing information base (RIB). The range of the interval is from 0 to 1200 seconds. The default value is 90 seconds.
	 Note The timer is only applicable for BGP dynamic neighbors. It is only set when BGP is restarted or is coming up for the first time for dynamic BGP neighbors.

To configure the maximum number of peers, use the following command in neighbor configuration mode:

Command	Purpose
maximum-peers <i>value</i> Example: switch(config-router-neighbor)# maximum-peers 120	Configures the maximum number of peers for this prefix peering. The range is from 1 to 1000.

This example shows how to configure a prefix peering that accepts up to 10 peers:

```
switch(config)# router bgp 65536
switch(config-router)# timers prefix-peer-timeout 120
```

```
switch(config-router)# neighbor 10.100.200.0/24 remote-as 65536
switch(config-router-neighbor)# maximum-peers 10
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)#
```

This example shows how to disable the peer prefix wait time:

```
switch(config)# router bgp 100
switch(config-router)# timers prefix-peer-wait 50
switch(config-router)#
```

Use the **show ip bgp neighbors** command to show the details of the configuration for that prefix peering with a list of the currently accepted instances and the counts of active, maximum concurrent, and total accepted peers.

Configuring BGP Authentication

You can configure BGP to authenticate route updates from peers using MD5 digests.

To configure BGP to use MD5 authentication, use the following command in neighbor configuration mode:

Command	Purpose
password [0 3 7] <i>string</i>	Configures an MD5 password for BGP neighbor sessions.
Example: switch(config-router-neighbor)# password BGPpassword	

Resetting a BGP Session

If you modify a route policy for BGP, you must reset the associated BGP peer sessions. If the BGP peers do not support route refresh, you can configure a soft reconfiguration for inbound policy changes. Cisco NX-OS automatically attempts a soft reset for the session.

To configure soft reconfiguration inbound, use the following command in neighbor address-family configuration mode:

Command	Purpose
soft-reconfiguration inbound always	Enables soft reconfiguration to store the inbound BGP route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Example: switch(config-router-neighbor-af)# soft-reconfiguration inbound always	

To reset a BGP neighbor session, use the following command in any mode:

Command	Purpose
clear bgp ip unicast <i>ip-address</i> soft { in out } Example: switch# clear bgp ip unicast 192.0.2.1 soft in	Resets the BGP session without tearing down the TCP session.

Modifying the Next-Hop Address

You can modify the next-hop address used in a route advertisement in the following ways:

- Disable the next-hop calculation and use the local BGP speaker address as the next-hop address.
- Set the next-hop address as a third-party address. Use this feature in situations where the original next-hop address is on the same subnet as the peer that the route is being sent to. Using this feature saves an extra hop during forwarding.

To modify the next-hop address, use the following parameters in commands address-family configuration mode:

Command	Purpose
next-hop-self Example: switch(config-router-neighbor-af) # next-hop-self	Uses the local BGP speaker address as the next-hop address in route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
next-hop-third-party Example: switch(config-router-neighbor-af) # next-hop-third-party	Sets the next-hop address as a third-party address. Use this command for single-hop EBGP peers that do not have next-hop-self configured.

Configuring BGP Next-Hop Address Tracking

BGP next-hop address tracking is enabled by default and cannot be disabled.

You can modify the delay interval between RIB checks to increase the performance of BGP next-hop tracking. You can configure the critical timer for routes that affect BGP next-hop reachability, and you can configure the noncritical timer for all other routes in the BGP table.

To modify the BGP next-hop address tracking, use the following commands address-family configuration mode:

Command	Purpose
nexthop trigger-delay {critical non-critical} <i>milliseconds</i> Example: switch(config-router-af)# nexthop trigger-delay critical 5000	Specifies the next-hop address tracking delay timer for critical next-hop reachability routes and for noncritical routes. The range is from 1 to 4294967295 milliseconds. The critical timer default is 3000. The noncritical timer default is 10000.
nexthop route-map <i>name</i> Example: switch(config-router-af)# nexthop route-map nextHopLimits	Specifies a route map to match the BGP next-hop addresses to. The name can be any case-sensitive, alphanumeric string up to 63 characters.

Configuring Next-Hop Filtering

BGP next-hop filtering allows you to specify that when a next-hop address is checked with the RIB, the underlying route for that next-hop address is passed through the route map. If the route map rejects the route, the next-hop address is treated as unreachable.

BGP marks all next-hops that are rejected by the route policy as invalid and does not calculate the best path for the routes that use the invalid next-hop address.

To configure BGP next-hop filtering, use the following command in address-family configuration mode:

Command	Purpose
nexthop route-map <i>name</i> Example: switch(config-router-af)# nexthop route-map nextHopLimits	Specifies a route map to match the BGP next-hop route to. The name can be any case-sensitive, alphanumeric string up to 63 characters.

Shrinking Next-Hop Groups When A Session Goes Down


You can configure BGP to shrink ECMP groups in an accelerated way when a session goes down.

This feature applies to the following BGP path failure events:

- Any single or multiple Layer 3 link failures
- BFD failure detections for BGP neighbors
- Administrative shutdown of BGP neighbors (using the shutdown command)

The accelerated handling of Layer 3 link failures is enabled by default and does not require a configuration command to be enabled.

To configure the accelerated handling of the last two events, use the following command in the router configuration mode:

Command	Purpose
neighbor-down fib-accelerate Example: <pre>switch(config-router)# neighbor-down fib-accelerate</pre>	Withdraws the corresponding next hop from all next-hop groups (ECMP groups and single next-hop routes) whenever a BGP session goes down.  Note This command applies to both IPv4 and IPv6 address-family routes and is supported only in a BGP-only environment where all non-direct routes are installed by BGP.

Disabling Capabilities Negotiation

You can disable capabilities negotiations to interoperate with older BGP peers that do not support capabilities negotiation.

To disable capabilities negotiation, use the following command in neighbor configuration mode:

Command	Purpose
dont-capability-negotiate Example: <pre>switch(config-router-neighbor)# dont-capability-negotiate</pre>	Disables capabilities negotiation. You must manually reset the BGP sessions after configuring this command.

Configuring eBGP

This section includes the following topics:

- [Configuring eBGP Next-Hop Unchanged, page 8-26](#)
- [Disabling eBGP Single-Hop Checking, page 8-26](#)
- [Configuring eBGP Multihop, page 8-27](#)
- [Configuring eBGP Routes in the Same Autonomous System, page 8-27](#)
- [Disabling a Fast External Failover, page 8-28](#)
- [Limiting the AS-path Attribute, page 8-28](#)
- [Configuring Local AS Support, page 8-28](#)

Configuring eBGP Next-Hop Unchanged

You can configure eBGP to send routes to an eBGP multihop peer without changing the next-hop address. By default, the device changes the next-hop address of a BGP route to its own address when the device sends out a route.

	Command	Purpose
Step 1	disable-connected-check Example: switch(config-router-neighbor)# disable-connected-check	Disables checking whether or not a single-hop eBGP peer is directly connected. You must manually reset the BGP sessions after using this command.
Step 2	configure terminal Example: switch# configure terminal	Enters global configuration mode.
Step 3	route-map <i>route-map name</i> Example: switch(config)# route-map route	Enters route map configuration mode.
Step 4	set ip next-hop unchanged Example: switch(config-route-map)# set ip next-hop unchanged	Configures the device to send BGP updates to the specified eBGP peer without modifying the next-hop address.
Step 5	exit Example: switch(config-route-map)# exit	Exits route map configuration mode.

This example shows how to set eBGP next-hop unchanged to send routes without changing the next-hop address:

```
switch# configure terminal
switch(config)# route-map route
switch(config-route-map)# set ip next-hop unchanged
switch(config-route-map)# exit
switch(config)#
```

Disabling eBGP Single-Hop Checking

You can configure eBGP to disable checking whether a single-hop eBGP peer is directly connected to the local router. Use this option for configuring a single-hop loopback eBGP session between directly connected switches.

To disable checking whether or not a single-hop eBGP peer is directly connected, use the following command in neighbor configuration mode:

Command	Purpose
disable-connected-check Example: switch(config-router-neighbor)# disable-connected-check	Disables checking whether or not a single-hop eBGP peer is directly connected. You must manually reset the BGP sessions after using this command.

Configuring eBGP Multihop

You can configure the eBGP time-to-live (TTL) value to support eBGP multihop. In some situations, an eBGP peer is not directly connected to another eBGP peer and requires multiple hops to reach the remote eBGP peer. You can configure the eBGP TTL value for a neighbor session to allow these multihop sessions.

To configure eBGP multihop, use the following command in neighbor configuration mode:

Command	Purpose
ebgp-multihop <i>ttl-value</i>	Configures the eBGP TTL value for eBGP multihop. The range is from 2 to 255. You must manually reset the BGP sessions after using this command.
Example: switch(config-router-neighbor)# ebgp-multihop 5	

Configuring eBGP Routes in the Same Autonomous System

You can configure eBGP learned routes to advertise between peers in the same autonomous system (AS).



Note

When route updates are sent between peers within the same AS number, they are dropped unless you enter the **allowas-in** command.

To disable AS peer checking, use the following command in neighbor configuration mode:

	Command	Purpose
Step 1	router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 64496	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.
Step 2	neighbor ipv4 remote-as <i>as-number</i> Example: switch(config-router)# neighbor 209.165.201.1 remote-as 64497	Configures the specified address type and AS number for a remote BGP peer. The <i>ip-address</i> format is x.x.x.x. The IPv6 <i>address-format</i> is A:B::C:D.
Step 3	address family ipv4 unicast Example: switch(config-router)# address family ip4 unicast	Enters neighbor address family configuration mode for the unicast specified address family.
Step 4	disable-peer-as-check Example: switch(config-router-neighbor-af)# disable-peer-as-check	Disables AS checking so that routes are updated between peers in the same AS.
Step 5	show bgp neighbor Example: switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors	(Optional) Displays information about BGP peers.

This example shows how to display BGP peer information:

```
switch(config)# show bgp neighbor 1.222.222.2
bgp neighbor is 1.222.222.2, remote as 2222, ebgp link, peer index 1
bgp version 4, remote router id 1.100.1.2 ####output truncated####
for address family:ipv4 unicast
bgp table version 54, neighbor version 54
3 accepted paths consume 108 bytes of memory
10 sent paths
peer asn check is disabled

####output omitted####
```

Disabling a Fast External Failover

Typically, when a BGP router loses connectivity to a directly connected eBGP peer, BGP triggers a fast external failover by resetting the eBGP session to the peer. You can disable this fast external failover to limit the instability caused by link flaps.

To disable a fast external failover, use the following command in router configuration mode:

Command	Purpose
no fast-external-failover Example: switch(config-router)# no fast-external-failover	Disables a fast external failover for eBGP peers. This command is enabled by default.

Limiting the AS-path Attribute

You can configure eBGP to discard routes that have a high number of AS numbers in the AS-path attribute.

To discard routes that have a high number of AS numbers in the AS-path attribute, use the following command in router configuration mode:

Command	Purpose
maxas-limit <i>number</i> Example: switch(config-router)# maxas-limit 50	Discards eBGP routes that have a number of AS-path segments that exceed the specified limit. The range is from 1 to 2000.

Configuring Local AS Support

The local AS feature allows a router to appear to be a member of a second autonomous system (AS), in addition to its real AS. Local AS allows two ISPs to merge without modifying peering arrangements. Routers in the merged ISP become members of the new autonomous system but continue to use their old AS numbers for their customers.

Local AS can only be used for true eBGP peers. You cannot use this feature for two peers that are members of different confederation subautonomous systems.

To configure eBGP local AS support, use the following command in neighbor configuration mode:

Command	Purpose
local-as <i>number</i> [no-prepend [replace-as [dual-as]]] Example: switch(config-router-neighbor)# local-as 1.1	<p>Configures eBGP to prepend the local AS <i>number</i> to the AS_PATH attribute.</p> <p>The local-as <i>number</i> 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.</p> <p>The no-prepend keyword ensures that the local-as <i>number</i> is not prepended to any downstream BGP neighbors except for the partner who is peering with the local-as number.</p> <p>The replace-as keyword ensures that only the local-as <i>number</i> of the peering session is prepended to the AS_PATH attribute. The autonomous-system number from the local BGP routing process is not prepended.</p> <p>The dual-as keyword configures the eBGP neighbor to establish a peering session using the real autonomous-system number (from the local BGP routing process) or by using the autonomous-system number configured as the Local AS).</p>

Configuring AS Confederations

To configure an AS confederation, you must specify a confederation identifier. The group of autonomous systems within the AS confederation looks like a single autonomous system with the confederation identifier as the autonomous system number.

To configure a BGP confederation identifier, use the following command in router configuration mode:

Command	Purpose
confederation identifier <i>as-number</i> Example: switch(config-router)# confederation identifier 64512	<p>Configures a confederation identifier for an AS confederation.</p> <p>Each confederation has a different sub-AS number, usually a private one (from 64512 to 65534).</p> <p>This command triggers an automatic notification and session reset for the BGP neighbor sessions.</p>

To configure the autonomous systems that belong to the AS confederation, use the following command in router configuration mode:

Command	Purpose
bgp confederation peers <i>as-number</i> [<i>as-number2...</i>]	Specifies a list of autonomous systems that belong to the confederation.
Example: switch(config-router)# bgp confederation peers 5 33 44	This command triggers an automatic notification and session reset for the BGP neighbor sessions.

Configuring an Autonomous System Path Containing Your Own Autonomous System

Enable the feature for BGP to accept the autonomous system (AS) path that contains your own AS.

BEFORE YOU BEGIN

Ensure that you have enabled the BGP feature (see the [“Enabling the BGP Feature”](#) section on page 7-11).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor ip-address remote-as** *as-number*
4. **address-family ipv4 unicast**
5. [**no** | **default**] **allowas-in** [*allowas-in-cnt*]
6. **end**
7. (Optional) **show running-config bgp**
8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	switch# configure terminal	Enters global configuration mode.
Step 2	switch(config)# router bgp <i>as-number</i>	Enters BGP mode and assigns the autonomous system number to the local BGP speaker. The <i>as-number</i> value range is from 1 to 65535.
Step 3	switch(config-router)# neighbor ip-address remote-as <i>as-number</i>	Enters neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 4	switch(config-router-neighbor)# address-family ipv4 unicast	Enters router address family configuration for the specified address family.

	Command	Purpose
Step 5	switch(config-router-neighbor-af)# [no default] allowas-in [allowas-in-cnt]	Enables the allowas-in feature for BGP and configures the number of occurrences of the AS number. For <i>allowas-in-cnt</i> , enter an integer between 1 and 10. By default, the number of occurrences of the AS number is set to 3.
Step 6	switch(config-router-neighbor-af)# end	Exits router address family configuration mode.
Step 7	switch# show running-config bgp	(Optional) Displays the BGP configuration.
Step 8	switch# copy running-config startup-config	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure the allowas-in feature for BGP and configure it for a unicast IPv4 address family:

```
switch# configure terminal
switch(config)# router bgp 77
switch(config-router)# neighbor 6.20.1.1 remote-as 66
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# allowas-in 5
switch(config-router-neighbor-af)# end
```

Configuring Route Reflector

You can configure iBGP peers as route reflector clients to the local BGP speaker, which acts as the route reflector. Together, a route reflector and its clients form a cluster. A cluster of clients usually has a single route reflector. In such instances, the cluster is identified by the router ID of the route reflector. To increase redundancy and avoid a single point of failure in the network, you can configure a cluster with more than one route reflector. You must configure all route reflectors in the cluster with the same 4-byte cluster ID so that a route reflector can recognize updates from route reflectors in the same cluster.

BEFORE YOU BEGIN

Ensure that you have enabled the BGP feature (see the [“Enabling the BGP Feature”](#) section on page 7-11).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **cluster-id** *cluster-id*
4. **address-family ipv4 unicast**
5. (Optional) **client-to-client reflection**
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **address-family ipv4 unicast**
9. **route-reflector-client**

10. **show bgp ip unicast neighbors**

11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp as-number Example: switch(config)# router bgp 65536 switch(config-router)#	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	cluster-id cluster-id Example: switch(config-router)# cluster-id 192.0.2.1	Configures the local router as one of the route reflectors that serve the cluster. You specify a cluster ID to identify the cluster. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 4	address-family ipv4 unicast Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	Enters router address family configuration mode for the specified address family.
Step 5	client-to-client reflection Example: switch(config-router-af)# client-to-client reflection	(Optional) Configures client-to-client route reflection. This feature is enabled by default. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 6	exit Example: switch(config-router-neighbor)# exit switch(config-router)#	Exits router address configuration mode.
Step 7	neighbor ip-address remote-as as-number Example: switch(config-router)# neighbor 192.0.2.10 remote-as 65536 switch(config-router-neighbor)#	Configures the IP address and AS number for a remote BGP peer.
Step 8	address-family ipv4 unicast Example: switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#	Enters neighbor address family configuration mode for the specified address family.
Step 9	route-reflector-client Example: switch(config-router-neighbor-af)# route-reflector-client	Configures the switch as a BGP route reflector and configures the neighbor as its client. This command triggers an automatic notification and session reset for the BGP neighbor sessions.

	Command or Action	Purpose
Step 10	show bgp ip unicast neighbors Example: switch(config-router-neighbor-af)# show bgp ip unicast neighbors	(Optional) Displays the BGP peers.
Step 11	copy running-config startup-config Example: switch(config-router-neighbor-af)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure the router as a route reflector and add one neighbor as a client:

```
switch(config)# router bgp 65536
switch(config-router)# neighbor 192.0.2.10 remote-as 65536
switch(config-router-neighbor)# address-family ip unicast
switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)# copy running-config startup-config
```

Configuring Route Dampening

You can configure route dampening to minimize route flaps propagating through your iBGP network.

To configure route dampening, use the following command in address-family or VRF address family configuration mode:

Command	Purpose
dampening [{ <i>half-life reuse-limit suppress-limit max-suppress-time</i> <i>route-map map-name</i> }] Example: switch(config-router-af)# dampening route-map bgpDamp	Disables capabilities negotiation. The parameter values are as follows: <ul style="list-style-type: none"> half-life—The range is from 1 to 45. reuse-limit—The range is from 1 to 20000. suppress-limit—The range is from 1 to 20000. max-suppress-time—The range is from 1 to 255.

Configuring Route Deletion Delay

When routes are deleted from the hardware faster than the BGP neighbor can remove them from the next-hop list, the traffic that is received for those routes is lost. You can delay route deletion from the hardware by configuring a route deletion interval.

This configuration is not applied when a subset of next hops is being removed from the route or when a backup path exists for the route. It applies only to a route that is completely getting deleted.

Command	Purpose
route delete dampen interval [seconds]	Delays route deletion from the hardware.
Example: switch(config)# route delete dampen interval 20 switch(config)#	The interval can range from 1 second to 30 seconds. The default value is 0 seconds.

Configuring Load Sharing and ECMP

You can configure the maximum number of paths that BGP adds to the route table for equal-cost multipath load balancing.

To configure the maximum number of paths, use the following command in router address-family configuration mode:

Command	Purpose
maximum-paths [ibgp] <i>maxpaths</i>	Configures the maximum number of equal-cost paths for load sharing. The range is from 1 to 16. The default is 1.
Example: switch(config-router-af)# maximum-paths 12	

Configuring Maximum Prefixes

You can configure the maximum number of prefixes that BGP can receive from a BGP peer. If the number of prefixes exceeds this value, you can optionally configure BGP to generate a warning message or tear down the BGP session to the peer.

To configure the maximum allowed prefixes for a BGP peer, use the following command in neighbor address-family configuration mode:

Command	Purpose
maximum-prefix <i>maximum</i> [threshold] [restart time warning-only]	Configures the maximum number of prefixes from a peer. The parameter ranges are as follows:
Example: switch(config-router-neighbor-af)# maximum-prefix 12	<ul style="list-style-type: none"> <i>maximum</i>—The range is from 1 to 300000. <i>Threshold</i>—The range is from 1 to 100 percent. The default is 75 percent. <i>time</i>—The range is from 1 to 65535 minutes. <p>This command triggers an automatic notification and session reset for the BGP neighbor sessions if the prefix limit is exceeded.</p>

Configuring Dynamic Capability

You can configure dynamic capability for a BGP peer.

To configure dynamic capability, use the following command in neighbor configuration mode:

Command	Purpose
dynamic-capability	Enables dynamic capability. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
Example: switch(config-router-neighbor) # dynamic-capability	This command is enabled by default.

Configuring Aggregate Addresses

You can configure aggregate address entries in the BGP route table.

To configure an aggregate address, use the following command in router address-family configuration mode:

Command	Purpose
aggregate-address <i>ip-prefix/length</i> [as-set] [summary-only] [advertise-map <i>map-name</i>] [attribute-map <i>map-name</i>] [suppress-map <i>map-name</i>]	Creates an aggregate address. The path advertised for this route is an autonomous system set that consists of all elements contained in all paths that are being summarized:
Example: switch(config-router-af) # aggregate-address 192.0.2.0/8 as-set	<ul style="list-style-type: none"> • The as-set keyword generates autonomous system set path information and community information from contributing paths. • The summary-only keyword filters all more specific routes from updates. • The advertise-map keyword and argument specify the route map used to select attribute information from selected routes. • The attribute-map keyword and argument specify the route map used to select attribute information from the aggregate. • The suppress-map keyword and argument conditionally filters more specific routes.

Suppressing BGP Routes

You can configure Cisco NX-OS to advertise newly learned BGP routes only after these routes are confirmed by the Forwarding Information Base (FIB) and programmed in the hardware. After the routes are programmed, subsequent changes to these routes do not require this hardware-programming check. BGP route suppression is not enabled by default.



Note

When you enable fib-suppression on the switch for routes that are not programmed locally in the hardware because of hardware table exhaustion, BGP advertises these failed routes even though they are not programmed locally in the hardware.

To suppress BGP routes, use the following command in the router configuration mode:

Command	Purpose
suppress-fib-pending	Suppresses newly learned BGP routes (IPv4 or IPv6) from being advertised to downstream BGP neighbors until the routes have been programmed in the hardware.
Example: <pre>switch(config-router)# suppress-fib-pending</pre>	

Configuring BGP Conditional Advertisement

You can configure BGP conditional advertisement to limit the routes that BGP propagates. You define the following two route maps:

- **Advertise map**—Specifies the conditions that the route must match before BGP considers the conditional advertisement. This route map can contain any appropriate match statements.
- **Exist map or nonexist map**—Defines the prefix that must exist in the BGP table before BGP propagates a route that matches the advertise map. The nonexist map defines the prefix that must not exist in the BGP table before BGP propagates a route that matches the advertise map. BGP processes only the permit statements in the prefix list match statements in these route maps.

If the route does not pass the condition, BGP withdraws the route if it exists in the BGP table.

BEFORE YOU BEGIN

Ensure that you have enabled the BGP feature (see the [“Enabling the BGP Feature”](#) section on page 7-11).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ipaddress* **remote-as** *as-number*
4. **address-family** **ipv4 unicast**
5. **advertise-map** *adv-map* {**exist-map** *exist-rmap* | **non-exist-map** *nonexist-rmap*}
6. (Optional) **show ip bgp neighbor**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>as-number</i> Example: switch(config)# router bgp 65536 switch(config-router)#	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: switch(config-router)# neighbor 192.168.1.2 remote-as 65537 switch(config-router-neighbor)#	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 4	address-family ipv4 unicast Example: switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#	Enters address family configuration mode.
Step 5	advertise-map <i>adv-map</i> { exist-map <i>exist-rmap</i> non-exist-map <i>nonexist-rmap</i> } Example: switch(config-router-neighbor-af)# advertise-map advertise exist-map exist	Configures BGP to conditionally advertise routes based on the two configured route maps: <ul style="list-style-type: none"> • adv-map—Specifies a route map with match statements that the route must pass before BGP passes the route to the next route map. The adv-map is a case-sensitive, alphanumeric string up to 63 characters. • exist-rmap—Specifies a route map with match statements for a prefix list. A prefix in the BGP table must match a prefix in the prefix list before BGP will advertise the route. The exist-rmap is a case-sensitive, alphanumeric string up to 63 characters. • nonexist-rmap—Specifies a route map with match statements for a prefix list. A prefix in the BGP table must not match a prefix in the prefix list before BGP will advertise the route. The nonexist-rmap is a case-sensitive, alphanumeric string up to 63 characters.

	Command	Purpose
Step 6	show ip bgp neighbor Example: switch(config-router-neighbor-af)# show ip bgp neighbor	(Optional) Displays information about BGP and the configured conditional advertisement route maps.
Step 7	copy running-config startup-config Example: switch(config-router-neighbor-af)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure BGP conditional advertisement:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# neighbor 192.0.2.2 remote-as 65537
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# advertise-map advertise exist-map exist
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# exit
switch(config-router)# exit
switch(config)# route-map advertise
switch(config-route-map)# match as-path pathList
switch(config-route-map)# exit
switch(config)# route-map exit
switch(config-route-map)# match ip address prefix-list plist
switch(config-route-map)# exit
switch(config)# ip prefix-list plist permit 209.165.201.0/27
```

Configuring Route Redistribution

You can configure BGP to accept routing information from another routing protocol and redistribute that information through the BGP network. Optionally, you can assign a default route for redistributed routes.

BEFORE YOU BEGIN

Ensure that you have enabled the BGP feature (see the [“Enabling the BGP Feature”](#) section on page 7-11).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **address-family ipv4 unicast**
4. **redistribute** {direct | {eigrp | ospf | ospfv3 | rip} *instance-tag* | static} **route-map** *map-name*
5. (Optional) **default-metric** *value*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>as-number</i> Example: switch(config)# router bgp 65536 switch(config-router)#	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	address-family ipv4 unicast Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	Enters address family configuration mode.
Step 4	redistribute { direct { eigrp ospf ospfv3 rip } instance-tag static } route-map <i>map-name</i> Example: switch(config-router-af)# redistribute eigrp 201 route-map Eigrpmap	Redistributes routes from other protocols into BGP. See the “ Configuring Route Maps ” section on page 16-12 for more information about route maps.
Step 5	default-metric <i>value</i> Example: switch(config-router-af)# default-metric 33	(Optional) Generates a default route into BGP.
Step 6	copy running-config startup-config Example: switch(config-router-af)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to redistribute EIGRP into BGP:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# redistribute eigrp 201 route-map Eigrpmap
switch(config-router-af)# copy running-config startup-config
```

Disabling BGP Dampening with Redistribution

When an IGP metric of routes redistributed into BGP changes, BGP has internal dampening that prevents an immediate route update to the BGP peers. It affects how BGP handles IGP metric changes reported for redistributed routes. BGP dampens these changes through a batch process with a 10-minute delay. This command enables you to adjust that delay or remove it altogether for a quicker response to these changes.

SUMMARY STEPS

1. **configure terminal**

2. **router bgp** *as-number*
3. **address-family ipv4 unicast**
4. **dampen-igp-metric** *seconds*

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>as-number</i> Example: switch(config)# router bgp 65536 switch(config-router)#	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	address-family ipv4 unicast Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	Enters address family configuration mode.
Step 4	dampen-igp-metric <i>seconds</i> Example: switch(config-router-af)# dampen-igp-metric 100	Configures dampening of IGP metric-related changes for redistributed routes.

This example shows how to configure BGP dampening for redistributed routes:

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# dampen-igp-metric 100
switch(config-router-af)#
```

Configuring Multiprotocol BGP

You can configure MP-BGP to support multiple address families, including IPv4 unicast routes.

BEFORE YOU BEGIN

Ensure that you have enabled the BGP feature (see the [“Enabling the BGP Feature”](#) section on page 7-11).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ip-address* **remote-as** *as-number*

4. **address-family ipv4 unicast**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>as-number</i> Example: switch(config)# router bgp 65536 switch(config-router)#	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: switch(config-router)# neighbor 192.168.1.2 remote-as 65537 switch(config-router-neighbor)#	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 4	address-family ipv4 unicast Example: switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#	Enters address family configuration mode.
Step 5	copy running-config startup-config Example: switch(config-router-neighbor-af)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to enable advertising and receiving IPv4 routes for unicast RPF for a neighbor:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ipv6 address 2001:0DB8::1
switch(config-if)# router bgp 65536
switch(config-router)# neighbor 192.168.1.2 remote-as 35537
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# exit
```

Configuring BGP Extended Community Site of Origin


To configure BGP extended community site of origin, use the following commands

Command	Purpose
router bgp <i>as-number</i> Example: switch(config)# router bgp 1 switch(config-router)#	Configures a BGP routing process and enters router configuration mode.
vrf <i>vrf-name</i> Example: switch(config-router)# vrf 450 switch(config-router-vrf)#	Enters the router VRF configuration mode and associates this BGP instance with a VRF.
neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: switch(config-router-vrf)# neighbor 1::1 remote-as 2 switch(config-router-vrf-neighbor)#	Configures the IP address and AS number for a remote BGP peer.
address-family { <i>ipv4</i> <i>ipv6</i> } unicast Example: switch(config-router-vrf-neighbor)# address-family ipv6 unicast switch(config-router-vrf-neighbor-af)#	Enters global address family configuration mode for the specified address family.
soo <i>value</i> Example: switch(config-router-vrf-neighbor-af)# soo 22:14	Configures the site of origin BGP extended community value. The value is in one of the following formats: –asn:number –IP address:number The number range is from 0 to 65535 for a 2-byte ASN or from 0 to 4294967295 for a 4-byte ASN.

Tuning BGP

You can tune BGP characteristics through a series of optional parameters.

To tune BGB, use the following optional commands in router configuration mode:


Command	Purpose
bestpath [always-compare-med compare-routerid med { missing-as-worst non-deterministic } as-path multipath-relax] Example: <pre>switch(config-router)# bestpath always-compare-med switch(config-router)# bestpath as-path multipath-relax</pre>	<p>Modifies the best-path algorithm. The optional parameters are as follows:</p> <ul style="list-style-type: none"> • always-compare-med—Compares MED on paths from different autonomous systems. • compare-routerid—Compares the router IDs for identical eBGP paths. • med missing-as-worst—Handles a missing MED as the highest MED. • med non-deterministic—Does not always select the best MED path from among the paths from the same autonomous system. • as-path multipath-relax—Allows the switch to handle the paths received from different AS numbers for multipath, if their AS-path lengths are the same and other multipath conditions are met.
enforce-first-as Example: <pre>switch(config-router)# enforce-first-as</pre>	<p>Enforces the neighbor autonomous system to be the first AS number listed in the AS_path attribute for eBGP.</p>
log-neighbor-changes Example: <pre>switch(config-router)# log-neighbor-changes</pre>	<p>Generates a system message when a neighbor changes state.</p> <div>  <p>Note To suppress neighbor status change messages for a specific neighbor, you can use the log-neighbor-changes disable command in neighbor configuration mode.</p> </div>

Command	Purpose
router-id <i>id</i> Example: switch(config-router)# router-id 209.165.20.1	Manually configures the router ID for this BGP speaker.
timers [bestpath-delay <i>delay</i> bgp <i>keepalive holdtime</i> prefix-peer-timeout <i>timeout</i>] Example: switch(config-router)# timers bgp 90 270	Sets the BGP timer values. The optional parameters are as follows: <ul style="list-style-type: none"> <i>delay</i>—Initial best-path timeout value after a restart. The range is from 0 to 3600 seconds. The default value is 300. <i>keepalive</i>—BGP session keepalive time. The range is from 0 to 3600 seconds. The default value is 60. <i>holdtime</i>—BGP session hold time. The range is from 0 to 3600 seconds. The default value is 180. <i>timeout</i>—Prefix peer timeout value. The range is from 0 to 1200 seconds. The default value is 30. You must manually reset the BGP sessions after configuring this command.

To tune BGP, use the following optional commands in router address-family configuration mode:

Command	Purpose
distance <i>ebgp-distance ibgp distance local-distance</i> Example: switch(config-router-af)# distance 20 100 200	Sets the administrative distance for BGP. The range is from 1 to 255. The defaults are as follows: <ul style="list-style-type: none"> eBGP distance—20. iBGP distance—200. local distance—220. Local distance is the administrative distance used for aggregate discard routes when they are installed in the RIB.
bestpath all-paths-ecmp Example: switch(config-router-af)# bestpath all-paths-ecmp	Treats all paths as ECMP during best path calculation

To tune BGP, use the following optional commands in neighbor configuration mode:

Command	Purpose
description <i>string</i> Example: switch(config-router-neighbor) # description main site	Sets a descriptive string for this BGP peer. The string can be up to 80 alphanumeric characters.
low-memory exempt Example: switch(config-router-neighbor) # low-memory exempt	Exempts this BGP neighbor from a possible shutdown due to a low memory condition.
transport connection-mode passive Example: switch(config-router-neighbor) # transport connection-mode passive	Allows a passive connection setup only. This BGP speaker does not initiate a TCP connection to a BGP peer. You must manually reset the BGP sessions after configuring this command.
remove-private-as Example: switch(config-router-neighbor) # remove-private-as	Removes private AS numbers from outbound route updates to an eBGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.  Note See the Guidelines and Limitations for Advanced BGP section for additional information on this command.
update-source <i>interface-type number</i> Example: switch(config-router-neighbor) # update-source ethernet 1/2	Configures the BGP speaker to use the source IP address of the configured interface for BGP sessions to the peer. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
log-neighbor-changes [disable] Example: switch(config-router-neighbor) # log-neighbor-changes disable	Generates a system message when this specific neighbor changes state. The disable option suppresses neighbor status change messages for this specific neighbor.

To tune BGP, use the following optional commands in neighbor address-family configuration mode:

Command	Purpose
suppress-inactive Example: switch(config-router-neighbor-af) # suppress-inactive	Advertises the best (active) routes only to the BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
default-originate [route-map <i>map-name</i>] Example: switch(config-router-neighbor-af) # default-originate	Generates a default route to the BGP peer.

Command	Purpose
filter-list <i>list-name</i> { in out } Example: switch(config-router-neighbor-af) # filter-list BGPFilter in	Applies an AS_path filter list to this BGP peer for inbound or outbound route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
prefix-list <i>list-name</i> { in out } Example: switch(config-router-neighbor-af) # prefix-list PrefixFilter in	Applies a prefix list to this BGP peer for inbound or outbound route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
send-community Example: switch(config-router-neighbor-af) # send-community	Sends the community attribute to this BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
send-extendedcommunity Example: switch(config-router-neighbor-af) # send-extendedcommunity	Sends the extended community attribute to this BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.

Configuring Virtualization

BEFORE YOU BEGIN

Ensure that you have enabled the BGP feature (see the [“Enabling the BGP Feature”](#) section on page 7-11).

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **exit**
4. **router bgp** *as-number*
5. **vrf** *vrf-name*
6. **neighbor** *ip-address* **remote-as** *as-number*
7. **bestpath as-path multipath-relax**
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	vrf context <i>vrf-name</i> Example: switch(config)# vrf context RemoteOfficeVRF switch(config-vrf)#	Creates a new VRF and enters VRF configuration mode.
Step 3	exit Example: switch(config-vrf)# exit switch(config)#	Exits VRF configuration mode.
Step 4	router bgp <i>as-number</i> Example: switch(config)# router bgp 65536 switch(config-router)#	Creates a new BGP process with the configured autonomous system number.
Step 5	vrf <i>vrf-name</i> Example: switch(config-router)# vrf RemoteOfficeVRF switch(config-router-vrf)#	Enters the router VRF configuration mode and associates this BGP instance with a VRF.
Step 6	neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: switch(config-router-vrf)# neighbor 209.165.201.1 remote-as 65536 switch(config-router--vrf-neighbor)#	Configures the IP address and AS number for a remote BGP peer.
Step 7	bestpath as-path multipath-relax Example: switch(config-router-vrf)# bestpath as-path multipath-relax	(Optional) Allows the switch to treat paths received from different AS's for multipath, if their AS-path lengths are the same and other multipath conditions are met.
Step 8	copy running-config startup-config Example: switch(config-router-vrf-neighbor)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to create a VRF and configure the router ID in the VRF:

```
switch# configure terminal
switch(config)# vrf context NewVRF
switch(config-vrf)# exit
switch(config)# router bgp 65536
switch(config-router)# vrf NewVRF
switch(config-router-vrf)# neighbor 209.165.201.1 remote-as 65536
switch(config-router-vrf-neighbor)# copy running-config startup-config
```

Verifying the Advanced BGP Configuration

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

Command	Purpose
show bgp all [summary] [vrf vrf-name]	Displays the BGP information for all address families.
show bgp convergence [vrf vrf-name]	Displays the BGP information for all address families.
show bgp ip unicast [ip-address] community {regex expression [community] [no-advertise] [no-export] [no-export-subconfed]} [vrf vrf-name]	Displays the BGP routes that match a BGP community.
show bgp [vrf vrf-name] ip unicast [ip-address] community-list list-name [vrf vrf-name]	Displays the BGP routes that match a BGP community list.
show bgp ip unicast [ip-address] extcommunity {regex expression generic [non-transitive transitive] aa4:nn [exact-match]} [vrf vrf-name]	Displays the BGP routes that match a BGP extended community.
show bgp ip unicast [ip-address] extcommunity-list list-name [exact-match] [vrf vrf-name]	Displays the BGP routes that match a BGP extended community list.
show bgp ip unicast [ip-address] {dampening dampened-paths [regex expression]} [vrf vrf-name]	Displays the information for BGP route dampening. Use the clear bgp dampening command to clear the route flap dampening information.
show bgp ip unicast [ip-address] history-paths [regex expression] [vrf vrf-name]	Displays the BGP route history paths.
show bgp ip unicast [ip-address] filter-list list-name [vrf vrf-name]	Displays the information for the BGP filter list.
show bgp ip unicast [ip-address] neighbors [ip-address] [vrf vrf-name]	Displays the information for BGP peers. Use the clear bgp neighbors command to clear these neighbors.
show bgp ip unicast [ip-address] {nexthop nexthop-database} [vrf vrf-name]	Displays the information for the BGP route next-hop.
show bgp paths	Displays the BGP path information.
show bgp ip unicast [ip-address] policy name [vrf vrf-name]	Displays the BGP policy information. Use the clear bgp policy command to clear the policy information.
show bgp ip unicast [ip-address] prefix-list list-name [vrf vrf-name]	Displays the BGP routes that match the prefix list.
show bgp ip unicast [ip-address] received-paths [vrf vrf-name]	Displays the BGP paths stored for soft reconfiguration.

Command	Purpose
show bgp ip unicast [<i>ip-address</i>] regexp <i>expression</i> [vrf <i>vrf-name</i>]	Displays the BGP routes that match the AS_path regular expression.
show bgp ip unicast [<i>ip-address</i>] route-map <i>map-name</i> [vrf <i>vrf-name</i>]	Displays the BGP routes that match the route map.
show bgp peer-policy <i>name</i> [vrf <i>vrf-name</i>]	Displays the information about BGP peer policies.
show bgp peer-session <i>name</i> [vrf <i>vrf-name</i>]	Displays the information about BGP peer sessions.
show bgp peer-template <i>name</i> [vrf <i>vrf-name</i>]	Displays the information about BGP peer templates. Use the clear bgp peer-template command to clear all neighbors in a peer template.
show bgp process	Displays the BGP process information.
show ip bgp <i>options</i>	Displays the BGP status and configuration information. This command has multiple options. See the <i>Cisco Nexus 3000 Series Command Reference</i> for more information.
show ip mbgp <i>options</i>	Displays the BGP status and configuration information. This command has multiple options. See the <i>Cisco Nexus 3000 Series Command Reference</i> for more information.
show running-configuration bgp	Displays the current running BGP configuration.

Displaying BGP Statistics

To display BGP statistics, use the following commands:

Command	Purpose
show bgp ip unicast [<i>ip-address</i>] flap-statistics [vrf <i>vrf-name</i>]	Displays the BGP route flap statistics. Use the clear bgp flap-statistics command to clear these statistics.
show bgp sessions [vrf <i>vrf-name</i>]	Displays the BGP sessions for all peers. Use the clear bgp sessions command to clear these statistics.
show bgp sessions [vrf <i>vrf-name</i>]	Displays the BGP sessions for all peers. Use the clear bgp sessions command to clear these statistics.
show bgp statistics	Displays the BGP statistics.

Related Topics

The following topics can give more information on BGP:

- [Chapter 8, “Configuring Advanced BGP”](#)
- [Chapter 16, “Configuring Route Policy Manager”](#)

Additional References

For additional information related to implementing BGP, see the following sections:

- [Related Documents, page 8-50](#)
- [MIBs, page 8-50](#)

Related Documents

Related Topic	Document Title
BGP CLI commands	<i>Cisco Nexus 3000 Series Command Reference</i>

MIBs

MIBs	MIBs Link
BGP4-MIB	To locate and download MIBs, go to the following URL:
CISCO-BGP4-MIB	http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml



Configuring BGP Additional Paths

This chapter describes how to configure additional paths for the Border Gateway Protocol (BGP).

This chapter includes the following sections:

- [Information About BGP Additional Paths, page 9-1](#)
- [Configuring BGP Additional Paths, page 9-4](#)
- [Configuration Examples for BGP Additional Paths, page 9-10](#)
- [Verifying the BGP Additional Paths Configuration, page 9-12](#)

Information About BGP Additional Paths

This section includes the following topics:

- [Overview, page 9-1](#)
- [Benefits of Additional BGP Paths, page 9-3](#)
- [BGP Additional Paths Functionality, page 9-3](#)

Overview

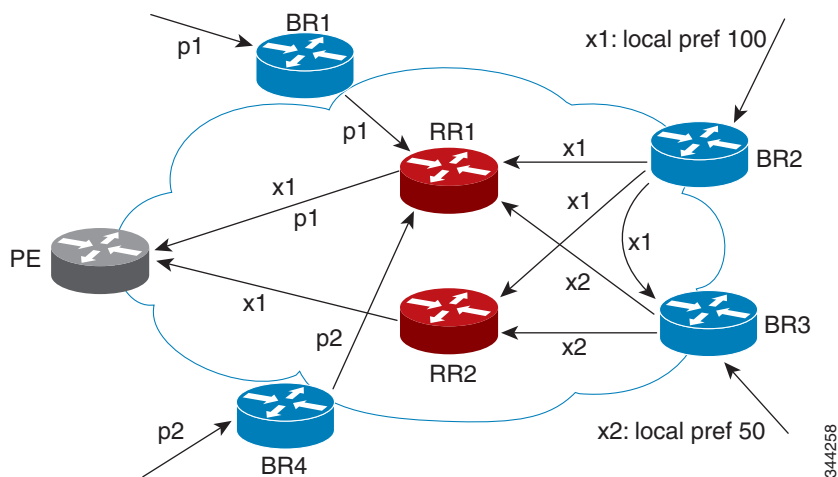
BGP routers and route reflectors (RRs) propagate only their best paths over their sessions. The advertisement of a prefix replaces the previous announcement of that prefix (this behavior is known as an implicit withdraw). The implicit withdraw can achieve better scaling, but at the cost of path diversity.

Path hiding can prevent efficient use of BGP multipath, prevent hitless planned maintenance, and lead to multi-exit discriminator (MED) oscillations and suboptimal hot-potato routing. In a next-hop failure, path hiding inhibits fast and local recovery because the network has to wait for BGP control plane convergence to restore traffic. The BGP Additional Paths feature offers path diversity; the best external or best internal features offer path diversity in limited scenarios.

The BGP Additional Paths feature allows multiple paths for the same prefix to be advertised without the new paths implicitly replacing the previous paths. Path diversity is achieved instead of path hiding.

Path-Hiding Scenario

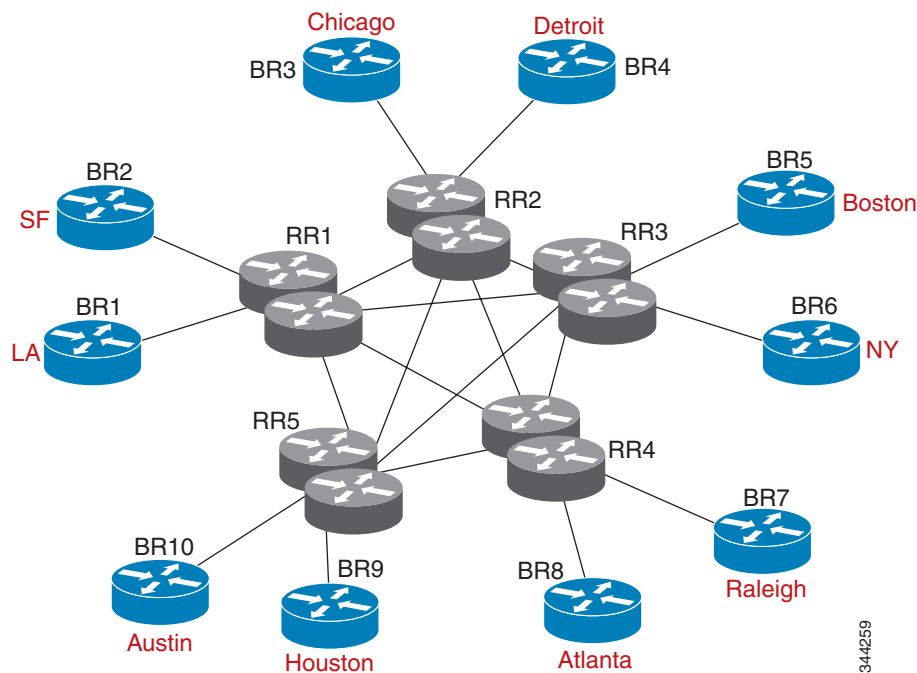
The following figure shows prefix p with paths p1 and p2 advertised from BR1 and BR4 to RR1. RR1 selects the best path of the two and then advertises only p1 to the PE.

Figure 9-1 RR Hiding an Additional Path

In the above figure, prefix x with path x1 is advertised from BR2 to BR3 (which has path x2) with local preference 100. BR3 also has path x2, but due to the routing policy, BR3 advertises to the RRs x1 (not shown) instead of x2, x2 is suppressed. You could turn on the advertisement of best external on BR3 and advertise x2 to the route reflectors (RRs), but, the RRs advertise only the best path.

Suboptimal Hot-Potato Routing Scenario

To minimize internal transport costs, transit Internet service providers try to forward packets to the closest exit point (according to the Interior Gateway Protocol (IGP) cost). This behavior is known as hot-potato routing. In the distributed RR cluster model of the figure below, assume traffic that is coming from LA must go to Mexico. All links have the same IGP cost. If there are two exit points toward Mexico—one toward Austin and one toward Atlanta—the border router will try to send traffic to Austin based on the lower IGP cost from LA toward Austin than toward Atlanta. In a centralized RR model where the central RR resides where RR3 is (and RR1, RR2, RR4, and RR5 do not exist), the closest exit point toward Mexico, as seen from RR3, might be Atlanta. Sending the traffic from LA toward the Atlanta border router (BR) results in suboptimal hot-potato routing and is not desirable.

Figure 9-2 *Distributed RR Cluster*

Benefits of Additional BGP Paths

BGP routers and route reflectors (RR) propagate only their best path over their sessions. The advertisement of a prefix replaces the previous announcement of that prefix (also known as an implicit withdraw).

While this behavior might achieve better scaling, it can prevent path diversity, which tends to be poor or completely lost. This behavior prevents efficient use of the BGP multipath, prevents hitless planned maintenance, and lead to multi-exit discriminator (MED) oscillations and suboptimal hot-potato routing. It also inhibits fast and local recovery for next-hop failures, because the network has to wait for BGP control plane convergence to restore traffic.

The BGP Additional Paths feature is a BGP extension that allows the advertisement of multiple paths for the same prefix without the new paths implicitly replacing any previous paths. This behavior promotes path diversity and reduces MED oscillations.

BGP Additional Paths Functionality

You can use the BGP Additional Paths feature by adding a path identifier to each path in the Network Layer Reachability Information (NLRI). The path identifier (ID) can be considered as something similar to a route distinguisher (RD) in virtual private networks (VPNs), except that a path ID can apply to any address family. Path IDs are unique to a peering session and are generated for each network. The path identifier is used to prevent a route announcement from implicitly withdrawing the previous one. The Additional Paths feature allows the advertisement of more paths, in addition to the best path and allows the advertisement of multiple paths for the same prefix, without the new paths implicitly replacing any previous paths.

The BGP Additional Paths feature requires you to take general steps:

1. Specify whether the device can send, receive, or send and receive additional paths at the address family level or the neighbor level. During session establishment, two BGP neighbors negotiate the additional path capabilities (whether they can send or receive) between them.
2. Select a set or sets of candidate paths for advertisement by specifying the selection criteria.
3. Advertise for a neighbor a set or sets of additional paths from the candidate paths marked.

To send or receive additional paths, the additional path capability must be negotiated between the neighbors. If no negotiation occurs, even if the selection criteria marks the best path and the neighbor is configured to advertise the marked paths, the selections are useless because only the best path is advertised.

Configuring BGP to send or receive additional paths triggers negotiation of an additional path's capability with the device's peers. Neighbors that have negotiated the capability are grouped together in an update group (if other update group policies allow), and in a separate update group from those peers that have not negotiated the capability. Therefore, the additional path capability causes the neighbor's update group membership to be recalculated.

Additional Path Selection

Only the best path is advertised to peers unless you configure the **set path-selection all advertise** command which advertises all BGP paths as additional paths to peers if the receive capability is enabled.

Advertising a Subset of the Paths Selected

Take care when you select a set of paths but want to advertise a different set of paths. If the set of paths you want to advertise is not a subset of the selected paths, you will not advertise the paths that you want advertised.

Guidelines and Limitations

Configuring BGP Additional Paths has the following guidelines and limitations:

- BGP add-path is not supported as a dynamic capability. It is included in the OPEN message, but not in CAPABILITY message. The configuration takes effect when the next session is established and does not cause established sessions to get torn down.

Configuring BGP Additional Paths

This section includes the following topics:

- [Configuring BGP Additional Paths for each Address Family, page 9-5](#)
- [Configuring BGP Additional Paths for each Neighbor, page 9-6](#)
- [Configuring Additional Paths Using a Peer Policy Template, page 9-7](#)
- [Filtering and Setting Actions for Additional Paths, page 9-9](#)

Configuring BGP Additional Paths for each Address Family

You can specify whether the device can send and receive additional paths to and from all neighbors within an address family.


BEFORE YOU BEGIN


Ensure that you have enabled the BGP feature.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **address family** {**ipv4** | **ipv6**} **unicast**
4. **additional-paths receive**
5. **additional-paths send**
6. **additional-paths selection route-map** *map-name*
7. **end**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>as-number</i> Example: switch(config)# router bgp 65000 switch(config-router)#	Enables BGP and assigns the autonomous system number to the local BGP speaker.
Step 3	address family { ipv4 ipv6 } unicast Example: switch(config-router)# address family ipv6 unicast	Enters address family configuration mode.
Step 4	additional-paths receive Example: switch(config-router-af)# additional-paths receive	(Optional) Enables BGP additional paths for a prefix to be received from a capable peer.  Note This capability applies to all neighbors under the specified address family unless the capability is explicitly disabled with the neighbor additional-paths receive disable command, which overrides the configuration for the address family.

	Command	Purpose
Step 5	additional-paths send Example: <pre>switch(config-router-af)# additional-paths send</pre>	(Optional) Enables BGP additional paths for a prefix to be sent to a capable peer.  Note This capability applies to all neighbors under the specified address family unless the capability is explicitly disabled with the neighbor additional-paths send disable command, which overrides the configuration for the address family.
Step 6	additional-paths selection route-map map-name Example: <pre>switch(config-router-af)# additional-paths selection route-map rmap</pre>	(Optional) Configures additional paths selection capability for a prefix.
Step 7	end Example: <pre>switch(config-router-af)# end</pre>	(Optional) Exits to privileged EXEC mode.

Configuring BGP Additional Paths for each Neighbor

You can configure whether a particular neighbor can send or receive additional paths.



BEFORE YOU BEGIN

Ensure that you have enabled the BGP feature (see the [“Enabling the BGP Feature”](#) section on page 7-11).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp as-number**
3. **neighbor {ipv4-address | ipv4-prefix/length | ipv6-address | ipv6-prefix/length} [remote-as {as-num} [.as-num]]**
4. **address family {ipv4 | ipv6} unicast**
5. **capability additional-paths receive [disable]**
6. **capability additional-paths send [disable]**
7. **end**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>as-number</i> Example: switch(config)# router bgp 65000 switch(config-router)#	Enables BGP and assigns the autonomous system number to the local BGP speaker.
Step 3	neighbor { <i>ipv4-address</i> <i>ipv4-prefix/length</i> <i>ipv6-address</i> <i>ipv6-prefix/length</i> } [remote-as { <i>as-num</i> } [. <i>as-num</i>]] Example: switch(config-router)# neighbor 2001:DB8::1037	Configures a BGP neighbor (router, VRF) and enters neighbor configuration mode.
Step 4	address family { <i>ipv4</i> <i>ipv6</i> } unicast Example: switch(config-router)# address family ipv6 unicast	Enters address family configuration mode.
Step 5	capability additional-paths receive [disable] Example: switch(config-router-af)# capability additional-paths receive	(Optional) Configures the receive additional paths capability for the specified neighbor.  Note This command overrides any send or receive capability that is configured at the address-family level.
Step 6	capability additional-paths send [disable] Example: switch(config-router-af)# capability additional-paths send	(Optional) Configures the send additional paths capability for the specified neighbor.  Note This command overrides any send or receive capability that is configured at the address-family level.
Step 7	end Example: switch(config-router-af)# end	(Optional) Exits to privileged EXEC mode.

Configuring Additional Paths Using a Peer Policy Template

You can send and receive additional paths by using a peer policy template.


BEFORE YOU BEGIN


Ensure that you have enabled the BGP feature (see the [“Enabling the BGP Feature”](#) section on page 7-11).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **template peer-policy** *template-name*
4. **capability additional-paths receive** [**disable**]
5. **capability additional-paths send** [**disable**]
6. **exit**
7. **neighbor** {*ipv4-address* | *ipv4-prefix/length* | *ipv6-address* | *ipv6-prefix/length*} [**remote-as** {*as-num*} [*.as-num*]]
8. **inherit peer-policy** *template-name sequence-number*
9. **end**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>as-number</i> Example: switch(config)# router bgp 65000	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	template peer-policy <i>template-name</i> Example: switch(config-router)# template peer-policy rr-client-ptl#	Enters policy-template configuration mode and creates a peer policy template.
Step 4	capability additional-paths receive [disable] Example: switch(config-router-af)# capability additional-paths receive	(Optional) Configures the receive additional paths capability for the specified neighbor. <div>  Note This command overrides any send or receive capability that is configured at the address-family level. </div>

	Command	Purpose
Step 5	capability additional-paths send [disable] Example: switch(config-router-af)# capability additional-paths send	(Optional) Configures the send additional paths capability for the specified neighbor.  Note This command overrides any send or receive capability that is configured at the address-family level.
Step 6	exit Example: switch(config-router-ptmp)# exit	Exits policy-template configuration mode and returns to router configuration mode.
Step 7	neighbor {ipv4-address ipv4-prefix/length ipv6-address ipv6-prefix/length} [remote-as {as-num} [.as-num]] Example: switch(config-router)# neighbor 2001:DB8::1037	Configures a BGP neighbor (router, VRF) and enters neighbor configuration mode.
Step 8	address-family ipv4 unicast Example: switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#	(Optional) Configures global address family configuration mode for the specified address family.
Step 9	inherit peer-policy template-name sequence-number Example: switch(config-router-neighbor-af)# inherit peer-policy rr-client-ptl 10	Sends a peer policy template to a neighbor so that the neighbor can inherit the configuration.
Step 10	end Example: switch(config-router-af)# end	(Optional) Exits to privileged EXEC mode.

Filtering and Setting Actions for Additional Paths

You can optionally use a route map to filter the paths to be advertised by matching on the prefix of additional paths that are candidates to be advertised. (These prefixes are configured with the **additional-paths selection** command.)

You can also optionally set one or more actions to take for those paths that pass through the route map. This procedure uses the **set metric** command. Other **set** commands are available that are not shown in this task.

You would set a metric for paths marked with **all** (all paths with a unique next-hop) if the neighbor is receiving the same routes from its neighbors. Suppose the neighbor 2001:DB8::1037 is receiving the same route from different neighbors. Routes received from the local device have a metric of 565 and routes from another device have a metric of 700. Routes with metric 565 have precedence over the routes with metric 700.

SUMMARY STEPS

- 1. **configure terminal**
- 2. **route-map** *route-name* [**deny** | **permit**] [*sequence-number*]
- 3. **set path-selection all advertise**
- 4. **set metric** *metric-value*
- 5. **end**

DETAILED STEPS

:

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	route-map <i>map-name</i> [deny permit] [<i>sequence-number</i>] Example: switch(config)# route-map add_path4 permit 10	Defines a route map and the conditions for redistributing routes from one routing protocol into another.
Step 3	set path-selection all advertise Example: switch(config-route-map)# set path-selection all advertise	Advertises all BGP paths as additional paths to peers if the receive capability is enabled.
Step 4	set metric <i>metric-value</i> Example: switch(config-route-map)# set metric 500	Sets the metric of the additional paths that pass the match criteria. <ul style="list-style-type: none">• Other set commands can be used to take action on the paths that pass the route map.
Step 5	end Example: switch(config-router-af)# end	(Optional) Exits to privileged EXEC mode.

Configuration Examples for BGP Additional Paths

- This section includes the following topics:
- [BGP Additional Paths Send and Receive Capabilities](#)
 - [BGP Additional Paths Using a Peer Policy Template](#)

BGP Additional Paths Send and Receive Capabilities

R1

In this example, R1's address is 2001:db8::1045; its neighbor R2 has an address of 2001:db8::1037. Updates are sent from R2 to R1 with additional-paths (all paths advertised). Updates are sent from R1 to R2 with only the classic BGP best path advertised because R2 can only send additional paths, not receive additional paths.

```
route-map add_path4 permit 10
set metric 500
set path-selection all advertise
!!
router bgp 1
address-family ipv6 unicast
additional-paths send
additional-paths receive
additional-paths selection route-map add_path4
neighbor 2001:db8::1037
address-family ipv6 unicast
capability additional-paths send
capability additional-paths receive
```

R2

```
route-map add_path4 permit 10
set metric 500
set path-selection all advertise
!!
router bgp 2
address-family ipv6 unicast
additional-paths selection route-map add_path4
neighbor 2001:db8::1045
address-family ipv6 unicast
capability additional-paths send
```

BGP Additional Paths Using a Peer Policy Template

This example shows that the neighbor with IP address 2001:db8::1037 has the send and receive capability for additional paths enabled through the template named rr-client-pt1:

```
router bgp 65000
```

```
address-family ipv6 unicast
additional-paths send
additional-paths receive
additional-paths selection route-map add_path4
neighbor 2001:db8::1037
address-family ipv6 unicast
inherit peer-policy rr-client-pt1 10
template peer-policy rr-client-pt1
capability additional-paths send
capability additional-paths receive
```

Verifying the BGP Additional Paths Configuration

To display information about the BGP additional paths configuration, perform the following tasks:

Command	Purpose
show ip bgp [<i>ip-address</i>]	Displays entries in the BGP table.
show ip bgp neighbors [<i>ip-address</i> [advertise-routes]]	Displays the configured neighbors and the other information specific to individual neighbor.



Configuring ECMP Recovery

This chapter describes how to configure the automatic recovery of equal-cost multipathing (ECMP) protocol on the Cisco NX-OS switch.

This chapter includes the following sections:

- [Information About ECMP Recovery, page 10-1](#)
- [Guidelines and Limitations for ECMP Recovery, page 10-1](#)
- [Default Settings, page 10-1](#)
- [Configuring ECMP Recovery, page 10-2](#)
- [Verifying ECMP Recovery Configuration, page 10-2](#)
- [Configuration Examples for ECMP Recovery, page 10-3](#)

Information About ECMP Recovery

The RIB adds ECMP routes to the FIB. When the hardware ECMP table becomes full, a single adjacency is created for all the subsequent ECMP routes. Until Cisco NX-OS Release 6.0(2)U2(2), when some entries in the ECMP table became free, these partially installed routes were not revisited to enable all paths.

Cisco NX-OS Release 6.0(2)U3(1) introduces the ECMP Recovery feature, which enables the FIB to automatically recover and reinstall partially installed ECMP routes when resources in the ECMP table become available. A threshold value can be configured for ECMP table resources to ensure that ECMP recovery is performed only when the configured percentage of ECMP table becomes free.

Guidelines and Limitations for ECMP Recovery

Only ECMP or VOBJ objects that are used by IPv4 and IPv6 routes are recovered and installed completely. Other features that use hardware ECMP objects, such as Policy Based Routing and VXLAN, are not supported by this feature.

Default Settings

ECMP recovery is disabled by default.

Configuring ECMP Recovery

To configure ECMP recovery, use the following command in global configuration mode:

SUMMARY STEPS

1. **configure terminal**
2. **hardware profile ecmp auto-recovery threshold *percentage***
3. (Optional) **copy running-config startup-config**
4. (Optional) **reload**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	hardware profile ecmp auto-recovery threshold <i>percentage</i> Example: switch(config)# hardware profile ecmp auto-recovery threshold 15	Configures the threshold for ECMP recovery. The threshold range is from 1 percent to 100 percent.
Step 3	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.
Step 4	reload Example: switch(config)# reload WARNING: This command will reboot the system Do you want to continue? (y/n) [n] y	(Optional) Reloads the Cisco Nexus 3000 Series switch software.

Verifying ECMP Recovery Configuration

To verify ECMP recovery configuration, use the following command in global configuration mode:

Command	Purpose
show running-config grep "hardware profile ecmp"	Displays the ECMP running configuration.

Configuration Examples for ECMP Recovery

This example shows how to configure ECMP recovery:

```
switch# configure terminal  
switch(config)# hardware profile ecmp auto-recovery threshold 15  
switch(config)# copy running-config startup-config  
switch(config)# reload
```

This example show how to verify ECMP recovery:

```
switch# configure terminal  
switch(config)# show running-config | grep "hardware profile ecmp"  
hardware profile ecmp auto-recovery threshold 15
```




Configuring ECMP for Host Routes

This chapter describes how to configure the equal-cost multipathing (ECMP) protocol for host routes on the Cisco NX-OS switch.

This chapter includes the following sections:

- [Information About ECMP for Host Routes, page 11-1](#)
- [Guidelines for ECMP for Host Routes, page 11-1](#)
- [Licensing Requirements for ECMP for Host Routes, page 11-2](#)
- [Prerequisites for ECMP for Host Routes, page 11-2](#)
- [Default Settings, page 11-2](#)
- [Configuring ECMP for Host Routes, page 11-2](#)
- [Verifying the ECMP for Host Routes Configuration, page 11-4](#)
- [Configuration Examples for ECMP for Host Routes, page 11-4](#)
- [Additional References, page 11-5](#)

Information About ECMP for Host Routes

When you enable ECMP support for host routes, all unicast host routes are programmed into the longest-prefix match algorithm (LPM) table. ECMP for host routes is provided in the switch hardware. You configure this feature in the CLI using the **hardware profile unicast enable-host-ecmp** command.



Note

Host entries are stored in the LPM routing table instead of the host table when ECMP is configured for IPv4 (/32) routes and IPv6 (/128) routes.

Guidelines for ECMP for Host Routes

After enabling or disabling ECMP for host routes by using the **[no] hardware profile unicast enable-host-ecmp** command, ensure that you do the following:

- Save the current configuration on the switch by using the **copy running-config startup-config** command
- Reload the switch by using the **reload** command so that the configuration can be applied.

Text Part Number:

Licensing Requirements for ECMP for Host Routes

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	ECMP for host routes requires 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. For a complete explanation of the Cisco NX-OS licensing scheme, see the <i>Cisco NX-OS Licensing Guide</i> .

Prerequisites for ECMP for Host Routes

ECMP for host routes has the following prerequisites:

- Before you use this command, we recommend that you disable Unicast Reverse Path Forwarding (URPF) globally on the switch using the **system urpf disable** command, and then save the configuration and reload the switch. Disabling URPF globally extends the LPM table size.

Default Settings

ECMP for host routes is disabled by default.

Configuring ECMP for Host Routes

This section includes the following topics:

- [Enabling the ECMP for Host Routes Feature, page 11-2](#)

Enabling the ECMP for Host Routes Feature

You can enable the ECMP for host routes feature.

SUMMARY STEPS

1. **configure terminal**
2. (Optional) **system urpf disable**
3. **hardware profile unicast enable-host-ecmp**
4. **copy running-config startup-config**
5. **reload**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	system urpf disable Example: switch(config)# system urpf disable	(Optional) Disables URPF globally on the switch.
Step 3	hardware profile unicast enable-host-ecmp Example: switch(config)# hardware profile unicast enable-host-ecmp	Enables ECMP for host routes globally on the switch.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	Saves this configuration change.
Step 5	reload Example: switch(config)# reload WARNING: This command will reboot the system Do you want to continue? (y/n) [n] y	Reloads the Cisco Nexus 3000 Series switch software.

Disabling the EMCP for Host Routes Feature

You can disable the ECMP for host routes feature.

SUMMARY STEPS

1. **configure terminal**
2. **no hardware profile unicast enable-host-ecmp**
3. **copy running-config startup-config**
4. **reload**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	no hardware profile unicast enable-host-ecmp Example: switch(config)# no hardware profile unicast enable-host-ecmp	Disables ECMP for host routes globally on the switch and removes all associated configuration.
Step 3	copy running-config startup-config Example: switch(config)# copy running-config startup-config	Saves this configuration change.
Step 4	reload Example: switch(config)# reload WARNING: This command will reboot the system Do you want to continue? (y/n) [n] y	Reloads the Cisco Nexus 3000 Series switch software.

Verifying the ECMP for Host Routes Configuration

To display the ECMP for host routes configuration information, perform one of the following tasks:

Command	Purpose
show hardware profile status	Displays the unicast entries in hardware tables.
show running-config	Displays the running system configuration.

Configuration Examples for ECMP for Host Routes

This example shows how to disable URPF and configure ECMP for host routes:

```
switch# configure terminal
switch(config)# system urpf disable
switch(config)# hardware profile unicast enable-host-ecmp
switch(config)# copy running-config startup-config
switch(config)# reload
```

This example show how to disable ECMP for host routes:

```
switch# configure terminal
switch(config)# no hardware profile unicast enable-host-ecmp
switch(config)# copy running-config startup-config
switch(config)# reload
```

Additional References

For additional information related to implementing ECMP for host routes, see the following sections:

- [Related Documents, page 11-5](#)

Related Documents

Related Topic	Document Title
ECMP for host routes CLI commands	<i>Cisco Nexus 3000 Series Command Reference</i>



Configuring RIP

This chapter describes how to configure the Routing Information Protocol (RIP).

This chapter includes the following sections:

- [Information About RIP, page 12-1](#)
- [Licensing Requirements for RIP, page 12-4](#)
- [Prerequisites for RIP, page 12-4](#)
- [Guidelines and Limitations, page 12-4](#)
- [Default Settings, page 12-4](#)
- [Configuring RIP, page 12-5](#)
- [Verifying the RIP Configuration, page 12-17](#)
- [Displaying RIP Statistics, page 12-17](#)
- [Configuration Examples for RIP, page 12-18](#)
- [Related Topics, page 12-18](#)
- [Additional References, page 12-18](#)

Information About RIP

This section includes the following topics:

- [RIP Overview, page 12-2](#)
- [RIPv2 Authentication, page 12-2](#)
- [Split Horizon, page 12-2](#)
- [Route Filtering, page 12-3](#)
- [Route Summarization, page 12-3](#)
- [Route Redistribution, page 12-3](#)
- [Load Balancing, page 12-3](#)
- [Virtualization Support, page 12-4](#)

RIP Overview

RIP uses User Datagram Protocol (UDP) data packets to exchange routing information in small internetworks. RIPv2 supports IPv4. RIPv2 uses an optional authentication feature supported by the RIPv2 protocol (see the [“RIPv2 Authentication” section on page 12-2](#)).

RIP uses the following two message types:

- Request—Sent to the address to request route updates from other RIP-enabled routers.
- Response—Sent every 30 seconds by default (see the [“Verifying the RIP Configuration” section on page 12-17](#)). The router also sends response messages after it receives a Request message. The response message contains the entire RIP route table. RIP sends multiple response packets for a request if the RIP routing table cannot fit in one response packet.

RIP uses a *hop count* for the routing metric. The hop count is the number of routers that a packet can traverse before reaching its destination. A directly connected network has a metric of 1; an unreachable network has a metric of 16. This small range of metrics makes RIP an unsuitable routing protocol for large networks.

RIPv2 Authentication

You can configure authentication on RIP messages to prevent unauthorized or invalid routing updates in your network. Cisco NX-OS supports a simple password or an MD5 authentication digest.

You can configure the RIP authentication per interface by using keychain management for the authentication keys. Keychain management allows you to control changes to the authentication keys used by an MD5 authentication digest or simple text password authentication.

To use an MD5 authentication digest, you configure a password that is shared at the local router and all remote RIP neighbors. Cisco NX-OS creates an MD5 one-way message digest based on the message itself and the encrypted password and sends this digest with the RIP message (Request or Response). The receiving RIP neighbor validates the digest by using the same encrypted password. If the message has not changed, the calculation is identical and the RIP message is considered valid.

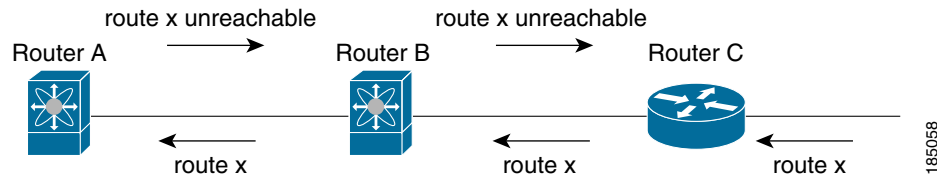
An MD5 authentication digest also includes a sequence number with each RIP message to ensure that no message is replayed in the network.

Split Horizon

You can use split horizon to ensure that RIP never advertises a route out of the interface where it was learned.

Split horizon is a method that controls the sending of RIP update and query packets. When you enable split horizon on an interface, Cisco NX-OS does not send update packets for destinations that were learned from this interface. Controlling update packets in this manner reduces the possibility of routing loops.

You can use split horizon with poison revers to configure an interface to advertise routes learned by RIP as unreachable over the interface that learned the routes. [Figure 12-1](#) shows a sample RIP network with split horizon with poison reverse enabled.

Figure 12-1 RIP with Split Horizon Poison Reverse

Router C learns about route X and advertises that route to router B. Router B in turn advertises route X to Router A, but sends a route X unreachable update back to router C.

By default, split horizon is enabled on all interfaces.

Route Filtering

You can configure a route policy on a RIP-enabled interface to filter the RIP updates. Cisco NX-OS updates the route table with only those routes that the route policy allows.

Route Summarization

You can configure multiple summary aggregate addresses for a specified interface. Route summarization simplifies route tables by replacing a number of more-specific addresses with an address that represents all the specific addresses. For example, you can replace 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24 with one summary address, 10.1.0.0/16.

If more specific routes are in the routing table, RIP advertises the summary address from the interface with a metric equal to the maximum metric of the more specific routes.



Note

Cisco NX-OS does not support automatic route summarization.

Route Redistribution

You can use RIP to redistribute static routes or routes from other protocols. When you configure redistribution use a route policy to control which routes are passed into RIP. A route policy allows you to filter routes based on attributes such as the destination, origination protocol, route type, route tag, and so on. For more information, see [Chapter 16, “Configuring Route Policy Manager.”](#)

Whenever you redistribute routes into a RIP routing domain, by default Cisco NX-OS does not redistribute the default route into the RIP routing domain. You can generate a *default route* into RIP, which can be controlled by a route policy.

You also configure the default metric that is used for all imported routes into RIP.

Load Balancing

You can use load balancing to allow a router to distribute traffic over all the router network ports that are the same distance from the destination address. Load balancing increases the utilization of network segments and increases effective network bandwidth.

Cisco NX-OS supports the Equal Cost Multiple Paths (ECMP) feature with up to 16 equal-cost paths in the RIP route table and the unicast RIB. You can configure RIP to load balance traffic across some or all of those paths.

Virtualization Support

Cisco NX-OS supports multiple instances of the RIP protocol that runs on the same system. RIP supports virtual routing and forwarding (VRF) instances .

By default, Cisco NX-OS places you in the default VRF unless you specifically configure another VRF. See [Chapter 14, “Configuring Layer 3 Virtualization.”](#)

Licensing Requirements for RIP

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	RIP requires 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. For a complete explanation of the Cisco NX-OS licensing scheme, see the <i>Cisco NX-OS Licensing Guide</i> .
	Note Make sure the LAN Base Services license is installed on the switch to enable Layer 3 interfaces.

Prerequisites for RIP

RIP has the following prerequisites:

- You must enable the RIP feature (see the [“Enabling the RIP Feature”](#) section on page 12-5).

Guidelines and Limitations

RIP has the following configuration guidelines and limitations:

- Cisco NX-OS does not support RIPv1. If Cisco NX-OS receives a RIPv1 packet, it logs a message and drops the packet.
- Cisco NX-OS does not establish adjacencies with RIPv1 routers.

Default Settings

[Table 12-1](#) lists the default settings for RIP parameters.

Table 12-1 **Default RIP Parameters**

Parameters	Default
Maximum paths for load balancing	16
RIP feature	Disabled
Split horizon	Enabled

Configuring RIP

This section includes the following topics:

- [Enabling the RIP Feature, page 12-5](#)
- [Creating a RIP Instance, page 12-6](#)
- [Restarting a RIP Instance, page 12-8](#)
- [Configuring RIP on an Interface, page 12-8](#)
- [Configuring RIP Authentication, page 12-9](#)
- [Configuring a Passive Interface, page 12-11](#)
- [Configuring Route Summarization, page 12-11](#)
- [Configuring Route Summarization, page 12-11](#)
- [Configuring Route Redistribution, page 12-12](#)
- [Configuring Virtualization, page 12-13](#)
- [Tuning RIP, page 12-16](#)

**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 the RIP Feature

You must enable the RIP feature before you can configure RIP.

SUMMARY STEPS

1. **configure terminal**
2. **feature rip**
3. (Optional) **show feature**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	feature rip Example: switch(config)# feature rip	Enables the RIP feature.
Step 3	show feature Example: switch(config)# show feature	(Optional) Displays enabled and disabled features.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

Use the **no feature rip** command to disable the RIP feature and remove all associated configuration.

Command	Purpose
no feature rip Example: switch(config)# no feature rip	Disables the RIP feature and removes all associated configuration.

Creating a RIP Instance

You can create a RIP instance and configure the address family for that instance.

BEFORE YOU BEGIN

Ensure that you have enabled the RIP feature (see the [“Enabling the RIP Feature”](#) section on page 12-5).

SUMMARY STEPS

1. **configure terminal**
2. **router rip** *instance-tag*
3. **address-family ipv4 unicast**
4. (Optional) **show ip rip** [**instance** *instance-tag*] [**vrf** *vrf-name*]
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router rip <i>instance-tag</i> Example: switch(config)# router RIP Enterprise switch(config-router)#	Creates a new RIP instance with the configured <i>instance-tag value</i> .
Step 3	address-family ipv4 unicast Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	Configures the address family for this RIP instance and enters address-family configuration mode.
Step 4	show ip rip [instance <i>instance-tag</i>] [vrf <i>vrf-name</i>] Example: switch(config-router-af)# show ip rip	(Optional) Displays a summary of RIP information for all RIP instances.
Step 5	copy running-config startup-config Example: switch(config-router-af)# copy running-config startup-config	(Optional) Saves this configuration change.

Use the **no router rip** command to remove the RIP instance and the associated configuration.

Command	Purpose
no router rip <i>instance-tag</i> Example: switch(config)# no router rip Enterprise	Deletes the RIP instance and all associated configuration.

**Note**

You must also remove any RIP commands configured in interface mode.

You can configure the following optional parameters for RIP in address-family configuration mode:

Command	Purpose
distance <i>value</i> Example: switch(config-router-af)# distance 30	Sets the administrative distance for RIP. The range is from 1 to 255. The default is 120. See the “Administrative Distance” section on page 1-7.
maximum-paths <i>number</i> Example: switch(config-router-af)# maximum-paths 6	Configures the maximum number of equal-cost paths that RIP maintains in the route table. The range is from 1 to 16. The default is 16.

This example shows how to create a RIP instance for IPv4 and set the number of equal-cost paths for load balancing:

```
switch# configure terminal
switch(config)# router rip Enterprise
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# max-paths 10
switch(config-router-af)# copy running-config startup-config
```

Restarting a RIP Instance

You can restart a RIP instance. This clears all neighbors for the instance.

To restart a RIP instance and remove all associated neighbors, use the following command:

Command	Purpose
restart rip <i>instance-tag</i>	Restarts the RIP instance and removes all neighbors.
Example: switch(config)# restart rip Enterprise	

Configuring RIP on an Interface

You can add an interface to a RIP instance.

BEFORE YOU BEGIN

Ensure that you have enabled the RIP feature (see the [“Enabling the RIP Feature”](#) section on page 12-5).

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **no switchport**
4. **ip router rip** *instance-tag*
5. (Optional) **show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot/port*] [**vrf** *vrf-name*] [**detail**]
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	ip router rip <i>instance-tag</i> Example: switch(config-if)# ip router rip Enterprise	Associates this interface with a RIP instance.
Step 5	show ip rip [<i>instance instance-tag</i>] interface [<i>interface-type slot/port</i>] [<i>vrf vrf-name</i>] [<i>detail</i>] Example: switch(config-if)# show ip rip Enterprise ethernet 1/2	(Optional) Displays RIP information for an interface.
Step 6	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to add the Ethernet 1/2 interface to a RIP instance:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# ip router rip Enterprise
switch(config)# copy running-config startup-config
```

Configuring RIP Authentication

You can configure authentication for RIP packets on an interface.

BEFORE YOU BEGIN

Ensure that you have enabled the RIP feature (see the [“Enabling the RIP Feature”](#) section on page 12-5).

Configure a keychain if necessary before enabling authentication.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **no switchport**
4. **ip rip authentication mode**{text | md5}
5. **ip rip authentication keychain** *key*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	ip rip authentication mode {text md5} Example: switch(config-if)# ip rip authentication mode md5	Sets the authentication type for RIP on this interface as cleartext or MD5 authentication digest.
Step 5	ip rip authentication keychain <i>key</i> Example: switch(config-if)# ip rip authentication keychain RIPKey	Configures the authentication key used for RIP on this interface.
Step 6	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to create a keychain and configure MD5 authentication on a RIP interface:

```
switch# configure terminal
switch(config)# key chain RIPKey
switch(config)# key-string myrip
switch(config)# accept-lifetime 00:00:00 Jan 01 2000 infinite
switch(config)# send-lifetime 00:00:00 Jan 01 2000 infinite
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# ip rip authentication mode md5
switch(config-if)# ip rip authentication keychain RIPKey
switch(config-if)# copy running-config startup-config
```


Configuring a Passive Interface

You can configure a RIP interface to receive routes but not send route updates by setting the interface to passive mode.

To configure a RIP interface in passive mode, use the following command in interface configuration mode:

Command	Purpose
ip rip passive-interface Example: switch(config-if)# ip rip passive-interface	Sets the interface into passive mode.

Configuring Split Horizon with Poison Reverse

You can configure an interface to advertise routes learned by RIP as unreachable over the interface that learned the routes by enabling poison reverse.

To configure split horizon with poison reverse on an interface, use the following command in interface configuration mode:

Command	Purpose
ip rip poison-reverse Example: switch(config-if)# ip rip poison-reverse	Enables split horizon with poison reverse. Split horizon with poison reverse is disabled by default.

Configuring Route Summarization

You can create aggregate addresses that are represented in the routing table by a summary address. Cisco NX-OS advertises the summary address metric that is the smallest metric of all the more-specific routes.

To configure a summary address on an interface, use the following command in interface configuration mode:

Command	Purpose
ip rip summary-address <i>ip-prefix/mask-len</i> Example: switch(config-if)# ip router rip summary-address 192.0.2.0/24	Configures a summary address for RIP for IPv4 addresses.

Configuring Route Redistribution

You can configure RIP to accept routing information from another routing protocol and redistribute that information through the RIP network. Redistributed routes can optionally be assigned a default route.

BEFORE YOU BEGIN

Ensure that you have enabled the RIP feature (see the [“Enabling the RIP Feature”](#) section on page 12-5).

Configure a route map before configuring redistribution. See the [“Configuring Route Maps”](#) section on page 16-12 for details on configuring route maps.

SUMMARY STEPS

1. **configure terminal**
2. **router rip** *instance-tag*
3. **address-family ipv4 unicast**
4. **redistribute** {**bgp** *as* | **direct** | **eigrp** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} **route-map** *map-name*
5. (Optional) **default-information originate** [**always**] [**route-map** *map-name*]
6. (Optional) **default-metric** *value*
7. (Optional) **show ip rip route** [{*ip-prefix* [**longer-prefixes** | **shorter-prefixes**]}] [**vrf** *vrf-name*] [**summary**]
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router rip <i>instance-tag</i> Example: switch(config)# router rip Enterprise switch(config-router)#	Creates a new RIP instance with the configured <i>instance-tag value</i> .
Step 3	address-family ipv4 unicast Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	Enters address-family configuration mode.
Step 4	redistribute { bgp <i>as</i> direct { eigrp ospf ospfv3 rip } <i>instance-tag</i> static } route-map <i>map-name</i> Example: switch(config-router-af)# redistribute eigrp 201 route-map RIPmap	Redistributes routes from other protocols into RIP. See the “Configuring Route Maps” section on page 16-12 for more information about route maps.

	Command	Purpose
Step 5	default-information originate [always] [route-map <i>map-name</i>] Example: switch(config-router-af)# default-information originate always	(Optional) Generates a default route into RIP, optionally controlled by a route map.
Step 6	default-metric <i>value</i> Example: switch(config-router-af)# default-metric 10	(Optional) Sets the default metric for all redistributed routes. The range is from 1 to 15. The default is 1.
Step 7	show ip rip route [<i>ip-prefix</i>] [longer-prefixes shorter-prefixes] [vrf <i>vrf-name</i>] [summary] Example: switch(config-router-af)# show ip rip route	(Optional) Shows the routes in RIP.
Step 8	copy running-config startup-config Example: switch(config-router-af)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to redistribute EIGRP into RIP:

```
switch# configure terminal
switch(config)# router rip Enterprise
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# redistribute eigrp 201 route-map RIPmap
switch(config-router-af)# copy running-config startup-config
```

Configuring Virtualization

You can create multiple VRFs and use the same or multiple RIP instances in each VRF. You assign a RIP interface to a VRF.



Note

Configure all other parameters for an interface after you configure the VRF for an interface. Configuring a VRF for an interface deletes all the configuration for that interface.

BEFORE YOU BEGIN

Ensure that you have enabled the RIP feature (see the [“Enabling the RIP Feature”](#) section on page 12-5).

SUMMARY STEPS

1. **configure terminal**
2. **vrf** *vrf-name*
3. **exit**
4. **router rip** *instance-tag*

5. **vrf context** *vrf_name*
6. (Optional) **address-family ipv4 unicast**
7. (Optional) **redistribute** { **bgp as** | **direct** | { **eigrp** | **ospf** | **ospfv3** | **rip** } *instance-tag* | **static** }
route-map *map-name*
8. **interface ethernet** *slot/port*
9. **no switchport**
10. **vrf member** *vrf-name*
11. **ip-address** *ip-prefix/length*
12. **ip router rip** *instance-tag*
13. (Optional) **show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot/port*] [**vrf** *vrf-name*]
14. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	vrf <i>vrf-name</i> Example: switch(config)# vrf RemoteOfficeVRF switch(config-vrf)#	Creates a new VRF.
Step 3	exit Example: switch(config-vrf)# exit switch(config)#	Exits VRF configuration mode.
Step 4	router rip <i>instance-tag</i> Example: switch(config)# router rip Enterprise switch(config-router)#	Creates a new RIP instance with the configured instance tag.
Step 5	vrf context <i>vrf-name</i> Example: switch(config)# vrf context RemoteOfficeVRF switch(config-vrf)#	Creates a new VRF and enters VRF configuration mode.
Step 6	address-family ipv4 unicast Example: switch(config-router-vrf)# address-family ipv4 unicast switch(config-router-vrf-af)#	(Optional) Configures the VRF address family for this RIP instance.

	Command	Purpose
Step 7	redistribute { bgp <i>as</i> direct { eigrp ospf ospfv3 rip } <i>instance-tag</i> static } route-map <i>map-name</i> Example: switch(config-router-vrf-af)# redistribute eigrp 201 route-map RIPmap	(Optional) Redistributes routes from other protocols into RIP. See the “Configuring Route Maps” section on page 16-12 for more information about route maps.
Step 8	interface ethernet <i>slot/port</i> Example: switch(config-router-vrf-af)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 9	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 10	vrf member <i>vrf-name</i> Example: switch(config-if)# vrf member RemoteOfficeVRF	Adds this interface to a VRF.
Step 11	ip address <i>ip-prefix/length</i> Example: switch(config-if)# ip address 192.0.2.1/16	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 12	ip router rip <i>instance-tag</i> Example: switch(config-if)# ip router rip Enterprise	Associates this interface with a RIP instance.
Step 13	show ip rip [instance <i>instance-tag</i>] interface [<i>interface-type slot/port</i>] [vrf <i>vrf-name</i>] Example: switch(config-if)# show ip rip Enterprise ethernet 1/2	(Optional) Displays RIP information for an interface in a VRF.
Step 14	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to create a VRF and add an interface to the VRF:

```
switch# configure terminal
switch(config)# vrf context RemoteOfficeVRF
switch(config-vrf)# exit
switch(config)# router rip Enterprise
switch(config-router)# vrf RemoteOfficeVRF
switch(config-router-vrf)# address-family ipv4 unicast
switch(config-router-vrf-af)# redistribute eigrp 201 route-map RIPmap
switch(config-router-vrf-af)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# vrf member RemoteOfficeVRF
switch(config-if)# ip address 192.0.2.1/16
switch(config-if)# ip router rip Enterprise
switch(config-if)# copy running-config startup-config
```

Tuning RIP

You can tune RIP to match your network requirements. RIP uses several timers that determine the frequency of routing updates, the length of time before a route becomes invalid, and other parameters. You can adjust these timers to tune routing protocol performance to better suit your internetwork needs.



Note

You must configure the same values for the RIP timers on all RIP-enabled routers in your network.

You can use the following optional commands in address-family configuration mode to tune RIP:

Command	Purpose
timers basic <i>update timeout holddown garbage-collection</i> Example: switch(config-router-af)# timers basic 40 120 120 100	Sets the RIP timers in seconds. The parameters are as follows: <ul style="list-style-type: none"> • update—The range is from 5 to any positive integer. The default is 30. • timeout—The time that Cisco NX-OS waits before declaring a route as invalid. If Cisco NX-OS does not receive route update information for this route before the timeout interval ends, Cisco NX-OS declares the route as invalid. The range is from 1 to any positive integer. The default is 180. • holddown—The time during which Cisco NX-OS ignores better route information for an invalid route. The range is from 0 to any positive integer. The default is 180. • garbage-collection—The time from when Cisco NX-OS marks a route as invalid until Cisco NX-OS removes the route from the routing table. The range is from 1 to any positive integer. The default is 120.

You can use the following optional commands in interface configuration mode to tune RIP:

Command	Purpose
ip rip metric-offset <i>value</i> Example: switch(config-if)# ip rip metric-offset 10	Adds a value to the metric for every router received on this interface. The range is from 1 to 15. The default is 1.
ip rip route-filter { prefix-list <i>list-name</i> route-map <i>map-name</i> } [in out] Example: switch(config-if)# ip rip route-filter route-map InputMap in	Specifies a route map to filter incoming or outgoing RIP updates.

Verifying the RIP Configuration

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

Command	Purpose
show ip rip instance [<i>instance-tag</i>] [vrf <i>vrf-name</i>]	Displays the status for an instance of RIP.
show ip rip [instance <i>instance-tag</i>] interface <i>slot/port</i> [detail [vrf <i>vrf-name</i>]]	Displays the RIP status for an interface.
show ip rip [instance <i>instance-tag</i>] neighbor [<i>interface-type number</i>] [vrf <i>vrf-name</i>]	Displays the RIP neighbor table.
show ip } rip [instance <i>instance-tag</i>] route [<i>ip-prefix/length</i> [longer-prefixes shorter--prefixes]] [summary] [vrf <i>vrf-name</i>]	Displays the RIP route table.
show running-configuration rip	Displays the current running RIP configuration.

Displaying RIP Statistics

To display the RIP statistics, use the following commands:

Command	Purpose
show ip rip [instance <i>instance-tag</i>] policy statistics redistribute { bgp <i>as</i> direct { eigrp ospf ospfv3 rip } <i>instance-tag</i> static } [vrf <i>vrf-name</i>]	Displays the RIP policy status.
show ip rip [instance <i>instance-tag</i>] statistics <i>interface-type number</i>] [vrf <i>vrf-name</i>]	Displays the RIP statistics.

Use the **clear ip rip policy** command to clear policy statistics.

Use the **clear ip rip statistics** command to clear RIP statistics.

Configuration Examples for RIP

This example creates the Enterprise RIP instance in a VRF and adds Ethernet interface 1/2 to this RIP instance. The example also configures authentication for Ethernet interface 1/2 and redistributes EIGRP into this RIP domain.

```
vrf context NewVRF
!
feature rip
router rip Enterprise
 vrf NewVRF
  address-family ip unicast
    redistribute eigrp 201 route-map RIPmap
    max-paths 10
!
interface ethernet 1/2
 no switchport
 vrf NewVRF
 ip address 192.0.2.1/16
 ip router rip Enterprise
 ip rip authentication mode md5
 ip rip authentication keychain RIPKey
```

Related Topics

See [Chapter 16, “Configuring Route Policy Manager”](#) for more information on route maps.

Additional References

For additional information related to implementing RIP, see the following sections:

- [Related Documents, page 12-18](#)
- [Standards, page 12-18](#)

Related Documents

Related Topic	Document Title
RIP CLI commands	<i>Cisco Nexus 3000 Series Command Reference</i>

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—



Configuring Static Routing

This chapter describes how to configure static routing on the switch.

This chapter includes the following sections:

- [Information About Static Routing, page 13-1](#)
- [Licensing Requirements for Static Routing, page 13-3](#)
- [Prerequisites for Static Routing, page 13-3](#)
- [Guidelines and Limitations, page 13-3](#)
- [Default Settings, page 13-4](#)
- [Configuring Static Routing, page 13-4](#)
- [Verifying the Static Routing Configuration, page 13-6](#)
- [Configuration Examples for Static Routing, page 13-6](#)
- [Additional References, page 13-7](#)

Information About Static Routing

Routers forward packets using either route information from route table entries that you manually configure or the route information that is calculated using dynamic routing algorithms.

Static routes, which define explicit paths between two routers, cannot be automatically updated; you must manually reconfigure static routes when network changes occur. Static routes use less bandwidth than dynamic routes. No CPU cycles are used to calculate and analyze routing updates.

You can supplement dynamic routes with static routes where appropriate. You can redistribute static routes into dynamic routing algorithms but you cannot redistribute routing information calculated by dynamic routing algorithms into the static routing table.

You should use static routes in environments where network traffic is predictable and where the network design is simple. You should not use static routes in large, constantly changing networks because static routes cannot react to network changes. Most networks use dynamic routes to communicate between routers but may have one or two static routes configured for special cases. Static routes are also useful for specifying a gateway of last resort (a default router to which all unroutable packets are sent).

This section includes the following topics:

- [Administrative Distance, page 13-2](#)
- [Directly Connected Static Routes, page 13-2](#)

- [Fully Specified Static Routes, page 13-2](#)
- [Floating Static Routes, page 13-2](#)
- [Remote Next-hops for Static Routes, page 13-3](#)
- [BFD, page 13-3](#)
- [Virtualization Support, page 13-3](#)

Administrative Distance

An administrative distance is the metric used by routers to choose the best path when there are two or more routes to the same destination from two different routing protocols. An administrative distance guides the selection of one routing protocol (or static route) over another, when more than one protocol adds the same route to the unicast routing table. Each routing protocol is prioritized in order of most to least reliable using an administrative distance value.

Static routes have a default administrative distance of 1. A router prefers a static route to a dynamic route because the router considers a route with a low number to be the shortest. If you want a dynamic route to override a static route, you can specify an administrative distance for the static route. For example, if you have two dynamic routes with an administrative distance of 120, you would specify an administrative distance that is greater than 120 for the static route if you want the dynamic route to override the static route.

Directly Connected Static Routes

You need to specify only the output interface (the interface on which all packets are sent to the destination network) in a directly connected static route. The router assumes the destination is directly attached to the output interface and the packet destination is used as the next-hop address. The next-hop can be an interface, only for point-to-point interfaces. For broadcast interfaces, the next-hop must be an IPv4 address.

Fully Specified Static Routes

You must specify the output interface (the interface on which all packets are sent to the destination network) and the next-hop address in a fully specified static route. You can use a fully specified static route when the output interface is a multi-access interface and you need to identify the next-hop address. The next-hop address must be directly attached to the specified output interface.

Floating Static Routes

A floating static route is a static route that the router uses to back up a dynamic route. You must configure a floating static route with a higher administrative distance than the dynamic route that it backs up. In this instance, the router prefers a dynamic route to a floating static route. You can use a floating static route as a replacement if the dynamic route is lost.

**Note**

By default, a router prefers a static route to a dynamic route because a static route has a smaller administrative distance than a dynamic route.

Remote Next-hops for Static Routes

You can specify the next-hop address of a neighboring router that is not directly connected to the router for static routes with remote (non-directly attached) next-hops. If a static route has remote next-hops during data-forwarding, the next-hops are recursively used in the unicast routing table to identify the corresponding directly attached next-hop(s) that have reachability to the remote next-hops.

BFD

This feature supports bidirectional forwarding detection (BFD). BFD is a detection protocol designed to provide fast forwarding-path failure detection times. 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.

BFD for BGP is supported on eBGP single-hop peers and iBGP single-hop peers. For iBGP single-hop peers using BFD, you must configure the update-source option in neighbor configuration mode. BFD is not supported on other iBGP peers or for multihop eBGP peers.

Virtualization Support

Static routes support virtual routing and forwarding (VRF) instances. By default, Cisco NX-OS places you in the default VRF unless you specifically configure another VRF. For more information, see [Chapter 14, “Configuring Layer 3 Virtualization.”](#)

Licensing Requirements for Static Routing

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	Static routing requires 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. For a complete explanation of the Cisco NX-OS licensing scheme, see the <i>Cisco NX-OS Licensing Guide</i> .
	Note Make sure the LAN Base Services license is installed on the switch to enable Layer 3 interfaces.

Prerequisites for Static Routing

Static routing has the following prerequisites:

- The next-hop address for a static route must be reachable or the static route will not be added to the unicast routing table.

Guidelines and Limitations

Static routing has the following configuration guidelines and limitations:

- You can specify an interface as the next-hop address for a static route only for point-to-point interfaces such as GRE tunnels.

Default Settings

Table 13-1 lists the default settings for static routing parameters.

Table 13-1 Default Static Routing Parameters

Parameters	Default
administrative distance	1
RIP feature	disabled

Configuring Static Routing

This section includes the following topics:

- [Configuring a Static Route, page 13-4](#)
- [Configuring Virtualization, page 13-5](#)



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.

Configuring a Static Route

You can configure a static route on the router.

SUMMARY STEPS

1. **configure terminal**
2. **ip route** { *ip-prefix* | *ip-addr ip-mask* } { [*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*] } [**tag** *tag-value* [*pref*]]
3. (Optional) **show ip static-route**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	ip route { <i>ip-prefix</i> <i>ip-addr ip-mask</i> } { <i>[next-hop nh-prefix]</i> [<i>interface</i> <i>next-hop nh-prefix</i>]} [tag <i>tag-value</i> [<i>pref</i>] Example: switch(config)# ip route 192.0.2.0/8 ethernet 1/2 192.0.2.4	Configures a static route and the interface for this static route. You can optionally configure the next-hop address. The <i>pref</i> value sets the administrative distance. The range is from 1 to 255. The default is 1.
Step 3	show ip static-route Example: switch(config)# show ip static-route	(Optional) Displays information about static routes.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure a static route:

```
switch# configure terminal
switch(config)# ip route 192.0.2.0/8 192.0.2.10
switch(config)# copy running-config startup-config
```

Use the **no ip static-route** command to remove the static route.

Configuring Virtualization

You can configure a static route in a VRF.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **ip route** {*ip-prefix* | *ip-addr ip-mask*} {*next-hop | nh-prefix | interface*} [**tag** *tag-value* [*pref*]]
4. (Optional) **show ip static-route vrf** *vrf-name*
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	vrf context <i>vrf-name</i> Example: switch(config)# vrf context StaticVrf	Creates a VRF and enters VRF configuration mode.
Step 3	ip route { <i>ip-prefix</i> <i>ip-addr ip-mask</i> } { <i>next-hop</i> <i>nh-prefix</i> <i>interface</i> } [tag <i>tag-value</i> [<i>pref</i>]] Example: switch(config-vrf)# ip route 192.0.2.0/8 ethernet 1/2	Configures a static route and the interface for this static route. You can optionally configure the next-hop address. The <i>pref</i> value sets the administrative distance. The range is from 1 to 255. The default is 1.
Step 4	show ip static-route <i>vrf vrf-name</i> Example: switch(config-vrf)# show ip static-route	(Optional) Displays information on static routes.
Step 5	copy running-config startup-config Example: switch(config-vrf)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure a static route:

```
switch# configure terminal
switch(config)# vrf context StaticVrf
switch(config-vrf)# ip route 192.0.2.0/8 192.0.2.10
switch(config-vrf)# copy running-config startup-config
```

Verifying the Static Routing Configuration

To display the static routing configuration information, use this command:

Command	Purpose
show ip static-route	Displays the configured static routes.

Configuration Examples for Static Routing

This example shows how to configure static routing:

```
configure terminal
ip route 192.0.2.0/8 192.0.2.10
copy running-config startup-config
```

Additional References

For additional information related to implementing static routing, see the following sections:

- [Related Documents, page 13-7](#)

Related Documents

Related Topic	Document Title
Static Routing CLI	<i>Cisco Nexus 3000 Series Command Reference</i>



Configuring Layer 3 Virtualization

This chapter describes how to configure Layer 3 virtualization.

This chapter includes the following sections:

- [Layer 3 Virtualization, page 14-1](#)
- [Licensing Requirements for VRFs, page 14-5](#)
- [Guidelines and Limitations for VRF, page 14-5](#)
- [Guidelines and Limitations for VRF-Lite, page 14-5](#)
- [Guidelines and Limitations for VRF Route Leaking, page 14-6](#)
- [Default Settings, page 14-6](#)
- [Configuring VRFs, page 14-6](#)
- [Configuring Non-recursive Static IP Route CLI in Default VRF, page 14-13](#)
- [Configuration Examples for VRF, page 14-19](#)
- [Additional References, page 14-21](#)
- [Additional References, page 14-21](#)

Layer 3 Virtualization

This section includes the following topics:

- [Overview of Layer 3 Virtualization, page 14-1](#)
- [VRF and Routing, page 14-2](#)
- [VRF-Lite, page 14-2](#)
- [VRF-Aware Services, page 14-3](#)

Overview of Layer 3 Virtualization

Cisco NX-OS supports virtual routing and forwarding instances (VRFs). Each VRF contains a separate address space with unicast route tables for IPv4 and makes routing decisions independent of any other VRF.

Each router has a default VRF and a management VRF. All Layer 3 interfaces and routing protocols exist in the default VRF until you assign them to another VRF. The mgmt0 interface exists in the management VRF. With the VRF-lite feature, the switch supports multiple VRFs in customer edge (CE) switches. VRF-lite allows a service provider to support two or more virtual private networks (VPNs) with overlapping IP addresses using one interface.

**Note**

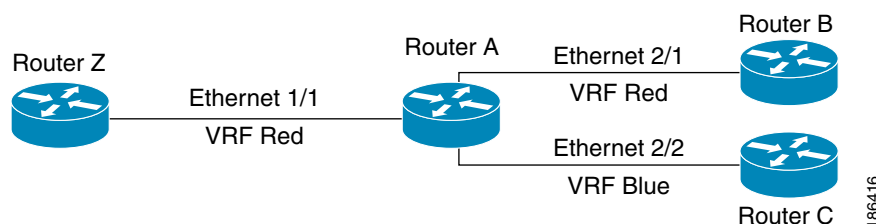
The switch does not use Multiprotocol Label Switching (MPLS) to support VPNs.

VRF and Routing

All unicast and multicast routing protocols support VRFs. When you configure a routing protocol in a VRF, you set routing parameters for the VRF that are independent of routing parameters in another VRF for the same routing protocol instance.

You can assign interfaces and route protocols to a VRF to create virtual Layer 3 networks. An interface exists in only one VRF. [Figure 14-1](#) shows one physical network split into two virtual networks with two VRFs. Routers Z, A, and B exist in VRF Red and form one address domain. These routers share route updates that do not include router C because router C is configured in a different VRF.

Figure 14-1 VRFs in a Network



By default, Cisco NX-OS uses the VRF of the incoming interface to select which routing table to use for a route lookup. You can configure a route policy to modify this behavior and set the VRF that Cisco NX-OS uses for incoming packets.

VRF supports route leaking (import or export) between VRFs. Certain limitations apply to route leaking in VRF-Lite. For more information, see [Guidelines and Limitations for VRF Route Leaking, page 14-6](#).

VRF-Lite

VRF-lite is a feature that enables a service provider to support two or more VPNs, where IP addresses can be overlapped among the VPNs. VRF-lite uses input interfaces to distinguish routes for different VPNs and forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each VRF. Interfaces in a VRF can be either physical, such as Ethernet ports, or logical, such as VLAN SVIs, but a Layer 3 interface cannot belong to more than one VRF at any time.

**Note**

Multiprotocol Label Switching (MPLS) and MPLS control plane are not supported in the VRF-lite implementation.

**Note**

VRF-lite interfaces must be Layer 3 interfaces.

VRF-Aware Services

A fundamental feature of the Cisco NX-OS architecture is that every IP-based feature is VRF aware.

The following VRF-aware services can select a particular VRF to reach a remote server or to filter information based on the selected VRF:

- AAA
- Call Home
- HSRP
- HTTP
- Licensing
- NTP
- RADIUS
- Ping and Traceroute
- SSH
- SNMP
- Syslog
- TACACS+
- TFTP
- VRRP

See the appropriate configuration guide for each service for more information on configuring VRF support in that service.

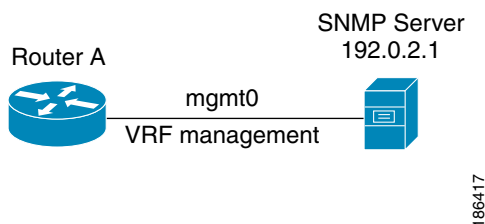
This section contains the following topics:

- [Reachability, page 14-3](#)
- [Filtering, page 14-4](#)
- [Combining Reachability and Filtering, page 14-4](#)

Reachability

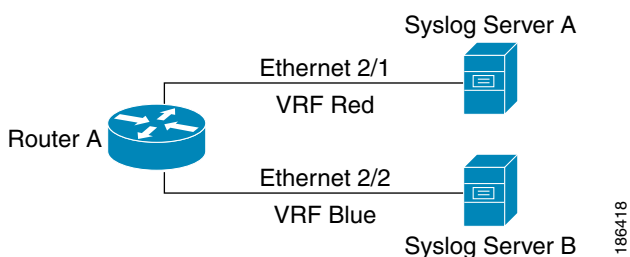
Reachability indicates which VRF contains the routing information necessary to get to the server providing the service. For example, you can configure an SNMP server that is reachable on the management VRF. When you configure that server address on the router, you also configure which VRF that Cisco NX-OS must use to reach the server.

[Figure 14-2](#) shows an SNMP server that is reachable over the management VRF. You configure router A to use the management VRF for SNMP server host 192.0.2.1.

Figure 14-2 Service VRF Reachability

Filtering

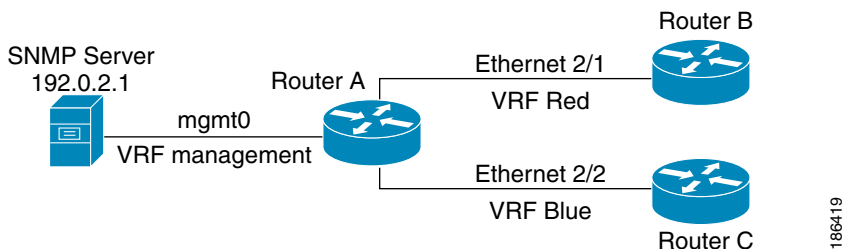
Filtering allows you to limit the type of information that goes to a VRF-aware service based on the VRF. For example, you can configure a syslog server to support a particular VRF. Figure 14-3 shows two syslog servers with each server supporting one VRF. syslog server A is configured in VRF Red, so Cisco NX-OS sends only system messages generated in VRF Red to syslog server A.

Figure 14-3 Service VRF Filtering

Combining Reachability and Filtering

You can combine reachability and filtering for VRF-aware services. You configure the VRF that Cisco NX-OS uses to connect to that service as well as the VRF that the service supports. If you configure a service in the default VRF, you can optionally configure the service to support all VRFs.

Figure 14-4 shows an SNMP server that is reachable on the management VRF. You can configure the SNMP server to support only the SNMP notifications from VRF Red, for example.

Figure 14-4 Service VRF Reachability Filtering

Licensing Requirements for VRFs

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	The default inband and outband management VRFs 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. For a complete explanation of the Cisco NX-OS licensing scheme, see the <i>Cisco NX-OS Licensing Guide</i> .
Cisco NX-OS	VRF-lite requires an Enterprise services license. For a complete explanation of the Cisco NX-OS licensing scheme, see the <i>Cisco NX-OS Licensing Guide</i> .

Guidelines and Limitations for VRF

VRFs have the following configuration guidelines and limitations in a VRF-lite scenario:

- When you make an interface a member of an existing VRF, Cisco NX-OS removes all Layer 3 configuration. You should configure all Layer 3 parameters after adding an interface to a VRF.
- You should add the mgmt0 interface to the management VRF and configure the mgmt0 IP address and other parameters after you add it to the management VRF.
- If you configure an interface for a VRF before the VRF exists, the interface is operationally down until you create the VRF.
- Cisco NX-OS creates the default and management VRFs by default. You should make the mgmt0 interface a member of the management VRF.
- The **write erase boot** command does not remove the management VRF configuration. You must use the **write erase** command and then the **write erase boot** command.

Guidelines and Limitations for VRF-Lite

VRF-lite has the following guidelines and limitations:

- A switch with VRF-lite has a separate IP routing table for each VRF, which is separate from the global routing table.
- Because VRF-lite uses different VRF tables, the same IP addresses can be reused. Overlapped IP addresses are allowed in different VPNs.
- VRF-lite does not support all MPLS-VRF functionality; it does not support label exchange, LDP adjacency, or labeled packets.
- Multiple virtual Layer 3 interfaces can be connected to a VRF-lite switch.
- The switch supports configuring a VRF by using physical ports, VLAN SVIs, or a combination of both. The SVIs can be connected through an access port or a trunk port.
- The Layer 3 TCAM resource is shared between all VRFs. To ensure that any one VRF has sufficient CAM space, use the **maximum routes** command.
- The total number of routes supported by all the VRFs is limited by the size of the TCAM.
- VRF-lite supports BGP, Routing Information Protocol (RIP), and static routing.
- VRF-lite does not support Enhanced Interior Gateway Routing Protocol (EIGRP).

- VRF-lite does not affect the packet switching rate.
- Multicast cannot be configured on the same Layer 3 interface at the same time.

Guidelines and Limitations for VRF Route Leaking

VRF route leaking has the following guidelines and limitations:

- Route leaking is supported between any two nondefault VRFs. It is also supported between the default VRF and any other VRF.
- Route leaking to the default VRF is not allowed because it is the global VRF.
- You can restrict route leaking to specific routes using route map filters to match designated IP addresses.
- By default, the maximum number of IP prefixes that can be leaked is set to 1000 routes. This number can be configured to any value from 0 to 1000.
- VRF route leaking requires an Enterprise license, and BGP must be enabled.
- A new CLI **default-information originate** is added to distribute the default static route to all BGP (VRF) Virtual Router Context and to install that route in the local BGP route table for all the BGP VRFs. For more information on the configuration, see [Configuring Virtualization, page 12-13](#).

Default Settings

[Table 14-1](#) lists the default settings for VRF parameters.

Table 14-1 *Default VRF Parameters*

Parameters	Default
Configured VRFs	default, management
routing context	default VRF

Configuring VRFs

This section contains the following topics:

- [Creating a VRF, page 14-7](#)
- [Assigning VRF Membership to an Interface, page 14-8](#)
- [Configuring VRF Parameters for a Routing Protocol, page 14-9](#)
- [Configuring a VRF-Aware Service, page 14-11](#)
- [Setting the VRF Scope, page 14-12](#)
- [Configuring Non-recursive Static IP Route CLI in Default VRF, page 14-13](#)
- [Configuring Recursive Static IP Route CLI in Default VRF, page 14-14](#)
- [Configuring Non-recursive Static IP Route CLI in Management VRF, page 14-14](#)
- [Configuring Recursive Static IP Route CLI in Management VRF, page 14-15](#)

- [Configuring Non-recursive Static IPv6 Route CLI in Default VRFs, page 14-16](#)
- [Configuring Recursive Static IPv6 Route CLI in Default VRF, page 14-16](#)
- [Configuring Non-recursive Static IPv6 Route CLI in Management VRF, page 14-17](#)
- [Configuring Recursive Static IPv6 Route CLI in Management VRF, page 14-18](#)

**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 VRF

You can create a VRF in a switch.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *name*
3. **ip route** { *ip-prefix* | *ip-addr ip-mask* } { [*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*] } [**tag** *tag-value* [*pref*]
4. (Optional) **show vrf** [*vrf-name*]
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	vrf context <i>name</i> Example: switch(config)# vrf definition Enterprise switch(config-vrf)#	Creates a new VRF and enters VRF configuration mode. The <i>name</i> can be any case-sensitive, alphanumeric string up to 32 characters.
Step 3	ip route { <i>ip-prefix</i> <i>ip-addr ip-mask</i> } { [<i>next-hop</i> <i>nh-prefix</i>] [<i>interface next-hop</i> <i>nh-prefix</i>] } [tag <i>tag-value</i> [<i>pref</i>] Example: switch(config-vrf)# ip route 192.0.2.0/8 ethernet 1/2 192.0.2.4	Configures a static route and the interface for this static route. You can optionally configure the next-hop address. The <i>preference</i> value sets the administrative distance. The range is from 1 to 255. The default is 1.

	Command	Purpose
Step 4	show vrf [<i>vrf-name</i>]	(Optional) Displays VRF information.
	Example: switch(config-vrf)# show vrf Enterprise	
Step 5	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

Use the **no vrf context** command to delete the VRF and the associated configuration:

Command	Purpose
no vrf context <i>name</i> Example: switch(config)# no vrf context Enterprise	Deletes the VRF and all associated configuration.

Any commands available in global configuration mode are also available in VRF configuration mode.

This example shows how to create a VRF and add a static route to the VRF:

```
switch# configure terminal
switch(config)# vrf context Enterprise
switch(config-vrf)# ip route 192.0.2.0/8 ethernet 1/2
switch(config-vrf)# exit
switch(config)# copy running-config startup-config
```

Assigning VRF Membership to an Interface

You can make an interface a member of a VRF.

BEFORE YOU BEGIN

Assign the IP address for an interface after you have configured the interface for a VRF.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrf member** *vrf-name*
4. **ip-address** *ip-prefix/length*
5. (Optional) **show vrf** *vrf-name interface interface-type number*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	vrf member <i>vrf-name</i> Example: switch(config-if)# vrf member RemoteOfficeVRF	Adds this interface to a VRF.
Step 4	ip address <i>ip-prefix/length</i> Example: switch(config-if)# ip address 192.0.2.1/16	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 5	show vrf <i>vrf-name interface interface-type number</i> Example: switch(config-vrf)# show vrf Enterprise interface ethernet 1/2	(Optional) Displays VRF information.
Step 6	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to add an interface to the VRF:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# vrf member RemoteOfficeVRF
switch(config-if)# ip address 192.0.2.1/16
switch(config-if)# copy running-config startup-config
```

Configuring VRF Parameters for a Routing Protocol

You can associate a routing protocol with one or more VRFs. See the appropriate chapter for information on how to configure VRFs for the routing protocol. This section uses OSPFv2 as an example protocol for the detailed configuration steps.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf** *instance-tag*

3. **vrf** *vrf-name*
4. (Optional) **maximum-paths** *paths*
5. **interface** *interface-type slot/port*
6. **vrf member** *vrf-name*
7. **ip address** *ip-prefix/length*
8. **ip router ospf** *instance-tag area area-id*
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf <i>instance-tag</i> Example: switch(config-vrf)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	vrf <i>vrf-name</i> Example: switch(config-router)# vrf RemoteOfficeVRF switch(config-router-vrf)#	Enters VRF configuration mode.
Step 4	maximum-paths <i>paths</i> Example: switch(config-router-vrf)# maximum-paths 4	(Optional) Configures the maximum number of equal OSPFv2 paths to a destination in the route table for this VRF. Used for load balancing.
Step 5	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 6	vrf member <i>vrf-name</i> Example: switch(config-if)# vrf member RemoteOfficeVRF	Adds this interface to a VRF.
Step 7	ip address <i>ip-prefix/length</i> Example: switch(config-if)# ip address 192.0.2.1/16	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.

	Command	Purpose
Step 8	ip router ospf instance-tag area area-id Example: switch(config-if)# ip router ospf 201 area 0	Assigns this interface to the OSPFv2 instance and area configured.
Step 9	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to create a VRF and add an interface to the VRF:

```
switch# configure terminal
switch(config)# vrf context RemoteOfficeVRF
switch(config-vrf)# exit
switch(config)# router ospf 201
switch(config-router)# vrf RemoteOfficeVRF
switch(config-router-vrf)# maximum-paths 4
switch(config-router-vrf)# interface ethernet 1/2
switch(config-if)# vrf member RemoteOfficeVRF
switch(config-if)# ip address 192.0.2.1/16
switch(config-if)# ip router ospf 201 area 0
switch(config-if)# exit
switch(config)# copy running-config startup-config
```

Configuring a VRF-Aware Service

You can configure a VRF-aware service for reachability and filtering. See the [“VRF-Aware Services” section on page 14-3](#) for links to the appropriate chapter or configuration guide for information on how to configure the service for VRFs. This section uses SNMP and IP domain lists as example services for the detailed configuration steps.

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server host ip-address [filter_vrf vrf-name] [use-vrf vrf-name]**
3. **vrf context [vrf-name]**
4. **ip domain-list domain-name [all-vrfs] [use-vrf vrf-name]**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	snmp-server host ip-address [filter-vrf vrf-name] [use-vrf vrf-name] Example: switch(config)# snmp-server host 192.0.2.1 use-vrf Red switch(config-vrf)#	Configures a global SNMP server and configures the VRF that Cisco NX-OS uses to reach the service. Use the filter-vrf keyword to filter information from the selected VRF to this server.
Step 3	vrf context vrf-name Example: switch(config)# vrf context Blue switch(config-vrf)#	Creates a new VRF.
Step 4	ip domain-list domain-name [all-vrfs] [use-vrf vrf-name] Example: switch(config-vrf)# ip domain-list List all-vrfs use-vrf Blue switch(config-vrf)#	Configures the domain list in the VRF and optionally configures the VRF that Cisco NX-OS uses to reach the domain name listed.
Step 5	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to send SNMP information for all VRFs to SNMP host 192.0.2.1, reachable on VRF Red:

```
switch# configure terminal
switch(config)# snmp-server host 192.0.2.1 for-all-vrfs use-vrf Red
switch(config)# copy running-config startup-config
```

This example shows how to Filter SNMP information for VRF Blue to SNMP host 192.0.2.12, reachable on VRF Red:

```
switch# configure terminal
switch(config)# vrf definition Blue
switch(config-vrf)# snmp-server host 192.0.2.12 use-vrf Red
switch(config)# copy running-config startup-config
```

Setting the VRF Scope

You can set the VRF scope for all EXEC commands (for example, **show** commands). This automatically restricts the scope of the output of EXEC commands to the configured VRF. You can override this scope by using the VRF keywords available for some EXEC commands.

To set the VRF scope, use the following command in EXEC mode:

Command	Purpose
routing-context vrf vrf-name Example: switch# routing-context vrf red switch%red#	Sets the routing context for all EXEC commands. Default routing context is the default VRF.

To return to the default VRF scope, use the following command in EXEC mode:

Command	Purpose
routing-context vrf default Example: switch%red# routing-context vrf default switch#	Sets the default routing context.

Configuring Non-recursive Static IP Route CLI in Default VRF

You can configure non-recursive static IP route CLI in the default VRF:

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	ip route 10.10.10.0/24 mgmt 0 10.197.121.1 vrf management	Configures the non-recursive static IP route CLI. It installs the route for 10.10.10.0/24 network in the default routing table. The nexthop 10.197.121.1 is in the management VRF.
Step 3	show ip route	Displays the output of the show ip route command.

See an example for the **show ip route** command output.

```
switch(config)# show ip route
```

```
IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.10.10.0/24, ubest/mbest: 1/0
    *via 10.197.121.1, mgmt0, [1/0], 00:00:57, static
27.1.1.0/24, ubest/mbest: 1/0, attached
    *via 27.1.1.1, Eth1/27, [0/0], 05:42:13, direct
27.1.1.1/32, ubest/mbest: 1/0, attached
    *via 27.1.1.1, Eth1/27, [0/0], 05:42:13, local
switch(config)#
```

Configuring Recursive Static IP Route CLI in Default VRF

You can configure recursive static IP route CLI in the default VRF:

	Command	Purpose
Step 1	#configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	(config)# ip route 20.20.20.0/24 10.197.121.1 vrf management	Configures the recursive static IP route CLI. It installs the route for 20.20.20.0/24 network in the default routing table. The nexthop 10.197.121.1 is in the management VRF.
Step 3	#show ip route	Displays the output of the show ip route command.

See an example for the **show ip route** command output.

```
switch(config)# show ip route

IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.10.10.0/24, ubest/mbest: 1/0
    *via 10.197.121.1, mgmt0, [1/0], 00:01:17, static
20.20.20.0/24, ubest/mbest: 1/0
    *via 10.197.121.1%management, [1/0], 00:00:03, static
27.1.1.0/24, ubest/mbest: 1/0, attached
    *via 27.1.1.1, Eth1/27, [0/0], 05:42:33, direct
27.1.1.1/32, ubest/mbest: 1/0, attached
    *via 27.1.1.1, Eth1/27, [0/0], 05:42:33, local
switch(config)#
```

Configuring Non-recursive Static IP Route CLI in Management VRF

You can configure non-recursive static IP route CLI in the management VRF:

	Command	Purpose
Step 1	#configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	(config)# vrf context management	Configures VRF context.

	Command	Purpose
Step 3	(config-vrf)# ip route 30.30.30.0/24 ethernet 1/1 1.1.1.2 vrf default	Configures the non-recursive static IP route CLI. It installs the route for 30.30.30.0/24 network in the management's routing table. The nexthop 1.1.1.2 is in the default vrf.
Step 4	#show ip route vrf management	Displays the output of the show ip route vrf management command.

See an example for the **show ip route** command output.

```
switch(config)# show ip route vrf management
IP Route Table for VRF "management"
'*' denotes best ucast next-hop
*** denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

0.0.0.0/0, ubest/mbest: 1/0
    *via 10.197.121.1, [1/0], 05:29:46, static
10.197.121.0/24, ubest/mbest: 1/0, attached
    *via 10.197.121.148, mgmt0, [0/0], 05:29:54, direct
10.197.121.148/32, ubest/mbest: 1/0, attached
    *via 10.197.121.148, mgmt0, [0/0], 05:29:54, local
30.30.30.0/24, ubest/mbest: 1/0
    *via 1.1.1.2, Eth1/1, [1/0], 00:00:06, static
switch(config)#
```

Configuring Recursive Static IP Route CLI in Management VRF

You can configure recursive static IP route CLI in the management VRF:

	Command	Purpose
Step 1	#configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	(config)# vrf context management	Configures VRF context.
Step 3	(config-vrf)# ip route 40.40.40.0/24 1.1.1.2 vrf default	Configures the non-recursive static IP route CLI. It installs the route for 40.40.40.0/24 network in the management's routing table. The nexthop 1.1.1.2 is in the default vrf.
Step 4	#show ip route vrf management	Displays the output of the show ip route vrf management command.

See an example for the **show ip route** command output.

```
switch(config)# show ip route vrf management
IP Route Table for VRF "management"
'*' denotes best ucast next-hop
*** denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
```

```
'%<string>' in via output denotes VRF <string>

0.0.0.0/0, ubest/mbest: 1/0
    *via 10.197.121.1, [1/0], 05:30:18, static
10.197.121.0/24, ubest/mbest: 1/0, attached
    *via 10.197.121.148, mgmt0, [0/0], 05:30:26, direct
10.197.121.148/32, ubest/mbest: 1/0, attached
    *via 10.197.121.148, mgmt0, [0/0], 05:30:26, local
30.30.30.0/24, ubest/mbest: 1/0
    *via 1.1.1.2, Eth1/1, [1/0], 00:00:38, static
40.40.40.0/24, ubest/mbest: 1/0
    *via 1.1.1.2%default, [1/0], 00:00:05, static
switch(config)#
```

Configuring Non-recursive Static IPv6 Route CLI in Default VRFs

You can configure non-recursive static IPv6 route CLI in the default VRF:

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	ipv6 route 3001::/64 mgmt 0 1001::2 vrf management	Configures the non-recursive static IPv6 route CLI.
Step 3	show ipv6 route	Displays the output of the show ipv6 route command.

See an example for the **show ipv6 route** command output.

```
switch(config)# show ipv6 route

IPv6 Routing Table for VRF "default"
'' denotes best ucast next-hop
''' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]

2700:1::/64, ubest/mbest: 1/0, attached
    *via 2700:1::1, Eth1/27, [0/0], 05:56:03, direct,
2700:1::1/128, ubest/mbest: 1/0, attached
    *via 2700:1::1, Eth1/27, [0/0], 05:56:03, local
3001::/64, ubest/mbest: 1/0
    *via 1001::2, mgmt0, [1/0], 00:00:04, static
```

Configuring Recursive Static IPv6 Route CLI in Default VRF

You can configure recursive static IPv6 route CLI in the default VRF:

	Command	Purpose
Step 1	#configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	(config)# ipv6 route 4001::/64 1001::2 vrf management	Configures the recursive static IPv6 route CLI.
Step 3	#show ipv6 route	Displays the output of the show ipv6 route command.

See an example for the **show ipv6 route** command output.

```
switch(config)# show ipv6 route
IPv6 Routing Table for VRF "default"
'*' denotes best ucast next-hop
*** denotes best mcast next-hop
'[x/y]' denotes [preference/metric]

2700:1::/64, ubest/mbest: 1/0, attached
    *via 2700:1::1, Eth1/27, [0/0], 05:56:26, direct,
2700:1::1/128, ubest/mbest: 1/0, attached
    *via 2700:1::1, Eth1/27, [0/0], 05:56:26, local
3001::/64, ubest/mbest: 1/0
    *via 1001:2, mgmt0, [1/0], 00:00:27, static
4001::/64, ubest/mbest: 1/0
    *via 1001:2%management, mgmt0, [1/0], 00:00:02, static
switch(config)#
```

Configuring Non-recursive Static IPv6 Route CLI in Management VRF

You can configure non-recursive static IPv6 route CLI in the management VRF:

	Command	Purpose
Step 1	#configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	(config)# vrf context management	Configures VRF context.
Step 3	(config-vrf)# ipv6 route 5001::/64 ethernet 1/1 2001::2 vrf default	Configures the non-recursive static IPv6 route CLI.
Step 4	#show ipv6 route vrf management	Displays the output of the show ipv6 route vrf management command.

See an example for the **show ip route** command output.

```
switch(config)# show ipv6 route vrf management
IPv6 Routing Table for VRF "management"
'*' denotes best ucast next-hop
```

```
''' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]

1001::/64, ubest/mbest: 1/0, attached
    *via 1001::1, mgmt0, [0/0], 00:02:19, direct,
1001::1/128, ubest/mbest: 1/0, attached
    *via 1001::1, mgmt0, [0/0], 00:02:19, local
5001::/64, ubest/mbest: 1/0
    *via 2001::2, Eth1/1, [1/0], 00:00:09, static
switch(config)#
```

Configuring Recursive Static IPv6 Route CLI in Management VRF

You can configure recursive static IPv6 route CLI in the management VRF:

	Command	Purpose
Step 1	#configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	(config)# vrf context management	Configures VRF context.
Step 3	(config-vrf)# ipv6 route 6001::/64 2001::2 vrf default	Configures the non-recursive static IPv6 route CLI.
Step 4	#show ipv6 route vrf management	Displays the output of the show ipv6 route vrf management command.

See an example for the **show ipv6 route** command output.

```
switch(config)# show ipv6 route vrf management
IPv6 Routing Table for VRF "management"
''' denotes best ucast next-hop
''' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]

1001::/64, ubest/mbest: 1/0, attached
    *via 1001::1, mgmt0, [0/0], 00:03:14, direct,
1001::1/128, ubest/mbest: 1/0, attached
    *via 1001::1, mgmt0, [0/0], 00:03:14, local
5001::/64, ubest/mbest: 1/0
    *via 2001::2, Eth1/1, [1/0], 00:01:04, static
6001::/64, ubest/mbest: 1/0
    *via 2001::2%default, Eth1/1, [1/0], 00:00:05, static
```

Verifying the VRF Configuration

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

Command	Purpose
show vrf [<i>vrf-name</i>]	Displays the information for all or one VRF.
show vrf [<i>vrf-name</i>] detail	Displays detailed information for all or one VRF.
show vrf [<i>vrf-name</i>] [interface <i>interface-type slot/port</i>]	Displays the VRF status for an interface.

Configuration Examples for VRF

This example shows how to configure VRF Red, add an SNMP server to that VRF, and add an instance of OSPF to VRF Red:

```
configure terminal
vrf context Red
  snmp-server host 192.0.2.12 use-vrf Red
router ospf 201
interface ethernet 1/2
  vrf member Red
  ip address 192.0.2.1/16
  ip router ospf 201 area 0
```

This example shows how to configure VRF Red and Blue, add an instance of OSPF to each VRF, and create an SNMP context for each OSPF instance in each VRF:

```
configure terminal
!Create the VRFs
vrf context Red
vrf context Blue
!Create the OSPF instances and associate them with each VRF
feature ospf
router ospf Lab
  vrf Red
router ospf Production
  vrf Blue
!Configure one interface to use ospf Lab on VRF Red
interface ethernet 1/2
  vrf member Red
  ip address 192.0.2.1/16
  ip router ospf Lab area 0
  no shutdown
!Configure another interface to use ospf Production on VRF Blue
interface ethernet 10/2
  vrf member Blue
  ip address 192.0.2.1/16
  ip router ospf Production area 0
  no shutdown
!configure the SNMP server
snmp-server user admin network-admin auth md5 nbv-12345
snmp-server community public ro
!Create the SNMP contexts for each VRF
snmp-server context lab instance Lab vrf Red
snmp-server context production instance Production vrf Blue
```

Use the SNMP context **lab** to access the OSPF-MIB values for the OSPF instance Lab in VRF Red in the previous example.

This example shows how to configure route leaking between two non-default VRF's, and from the default VRF to a non-default VRF:

```
feature bgp
vrf context Green
ip route 33.33.33.33/32 35.35.1.254
  address-family ipv4 unicast
    route-target import 3:3
    route-target export 2:2
    export map test
    import map test
    import vrf default map test

interface Ethernet1/7
vrf member Green
ip address 35.35.1.2/24
vrf context Shared
ip route 44.44.44.44/32 45.45.1.254
  address-family ipv4 unicast
    route-target import 1:1
    route-target import 2:2
    route-target export 3:3
    export map test
    import map test
    import vrf default map test

interface Ethernet1/11
vrf member Shared
ip address 45.45.1.2/24

router bgp 100
  address-family ipv4 unicast
    redistribute static route-map test
  vrf Green
    address-family ipv4 unicast
      redistribute static route-map test
  vrf Shared
    address-family ipv4 unicast
      redistribute static route-map test

ip prefix-list test seq 5 permit 0.0.0.0/0 le 32

route-map test permit 10
  match ip address prefix-list test

ip route 100.100.100.100/32 55.55.55.1

nexus# show ip route vrf all
IP Route Table for VRF "default"
'*' denotes best ucast next-hop
*** denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

55.55.55.0/24, ubest/mbest: 1/0, attached
  *via 55.55.55.5, Lo0, [0/0], 00:07:59, direct
55.55.55.5/32, ubest/mbest: 1/0, attached
  *via 55.55.55.5, Lo0, [0/0], 00:07:59, local
100.100.100.100/32, ubest/mbest: 1/0
  *via 55.55.55.1, [1/0], 00:07:42, static

IP Route Table for VRF "management"
'*' denotes best ucast next-hop
*** denotes best mcast next-hop
```

```
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

0.0.0.0/0, ubest/mbest: 1/0
  *via 10.29.176.1, [1/0], 12:53:54, static
10.29.176.0/24, ubest/mbest: 1/0, attached
  *via 10.29.176.233, mgmt0, [0/0], 13:11:57, direct
10.29.176.233/32, ubest/mbest: 1/0, attached
  *via 10.29.176.233, mgmt0, [0/0], 13:11:57, local

IP Route Table for VRF "Green"
'*' denotes best ucast next-hop
*** denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

33.33.33.33/32, ubest/mbest: 1/0
  *via 35.35.1.254, [1/0], 00:23:44, static
35.35.1.0/24, ubest/mbest: 1/0, attached
  *via 35.35.1.2, Eth1/7, [0/0], 00:26:46, direct
35.35.1.2/32, ubest/mbest: 1/0, attached
  *via 35.35.1.2, Eth1/7, [0/0], 00:26:46, local
44.44.44.44/32, ubest/mbest: 1/0
  *via 45.45.1.254%Shared, [20/0], 00:12:08, bgp-100, external, tag 100
100.100.100.100/32, ubest/mbest: 1/0
  *via 55.55.55.1%default, [20/0], 00:07:41, bgp-100, external, tag 100

IP Route Table for VRF "Shared"
'*' denotes best ucast next-hop
*** denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

33.33.33.33/32, ubest/mbest: 1/0
  *via 35.35.1.254%Green, [20/0], 00:12:34, bgp-100, external, tag 100
44.44.44.44/32, ubest/mbest: 1/0
  *via 45.45.1.254, [1/0], 00:23:16, static
45.45.1.0/24, ubest/mbest: 1/0, attached
  *via 45.45.1.2, Eth1/11, [0/0], 00:25:53, direct
45.45.1.2/32, ubest/mbest: 1/0, attached
  *via 45.45.1.2, Eth1/11, [0/0], 00:25:53, local
100.100.100.100/32, ubest/mbest: 1/0
  *via 55.55.55.1%default, [20/0], 00:07:41, bgp-100, external, tag 100
nexus(config)#
```

Additional References

For additional information related to implementing virtualization, see the following sections:

- [Related Documents, page 14-21](#)
- [Standards, page 14-22](#)

Related Documents

Related Topic	Document Title
VRF CLI	<i>Cisco Nexus 3000 Series Command Reference</i>

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—



Configuring the Unicast RIB and FIB

This chapter describes how to manage routes in the unicast Routing Information Base (RIB) and the Forwarding Information Base (FIB) on the Cisco NX-OS switch.

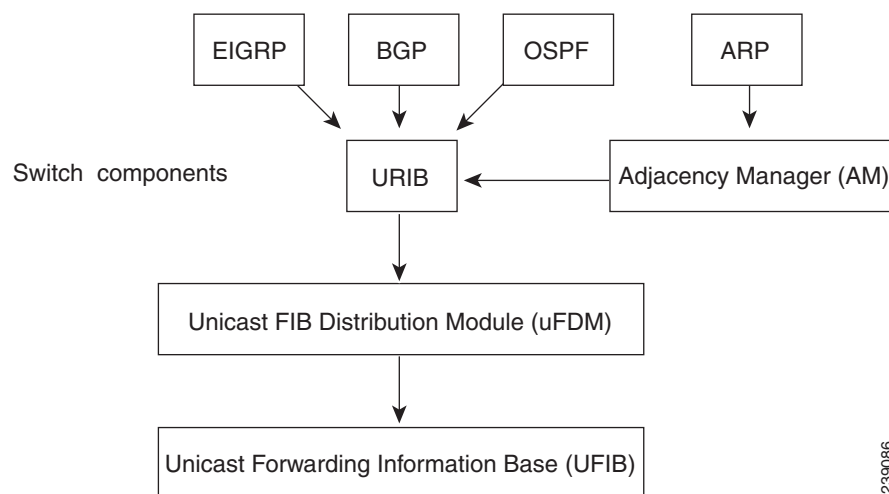
This chapter includes the following sections:

- Information About the Unicast RIB and FIB, page 15-1
- Licensing Requirements for the Unicast RIB and FIB, page 15-3
- Configuring the Unicast RIB and FIB, page 15-3
- Verifying the Unicast RIB and FIB Configuration, page 15-10
- Additional References, page 15-11

Information About the Unicast RIB and FIB

The unicast RIB (IPv4 RIB) and FIB are part of the Cisco NX-OS forwarding architecture, as shown in Figure 15-1.

Figure 15-1 Cisco NX-OS Forwarding Architecture



Text Part Number:

The unicast RIB maintains the routing table with directly connected routes, static routes, and routes learned from dynamic unicast routing protocols. The unicast RIB also collects adjacency information from sources such as the Address Resolution Protocol (ARP). The unicast RIB determines the best next-hop for a given route and populates the unicast forwarding information base (FIBs) by using the services of the unicast FIB distribution module (FDM).

Each dynamic routing protocol must update the unicast RIB for any route that has timed out. The unicast RIB then deletes that route and recalculates the best next-hop for that route (if an alternate path is available).

This section includes the following topics:

- Layer 3 Consistency Checker, page 15-2
- [FIB Tables, page 15-2](#)
- Virtualization Support, page 15-3

Layer 3 Consistency Checker

In rare instances, an inconsistency can occur between the unicast RIB and the FIB on each module. Cisco NX-OS supports the Layer 3 consistency checker. This feature detects inconsistencies between the unicast IPv4 RIB and the FIB on each interface module. Inconsistencies include the following:

- Missing prefix
- Extra prefix
- Wrong next-hop address
- Incorrect Layer 2 rewrite string in the ARP or neighbor discovery (ND) cache

The Layer 3 consistency checker compares the FIB entries to the latest adjacency information from the Adjacency Manager (AM) and logs any inconsistencies. The consistency checker then compares the unicast RIB prefixes to the module FIB and logs any inconsistencies. See the “Triggering the Layer 3 Consistency Checker” section on page 15-8.

You can then manually clear any inconsistencies. See the “Clearing Forwarding Information in the FIB” section on page 15-9.

FIB Tables

The hardware provides two tables, a TCAM table and a Hash table. The TCAM table is shared between the longest prefix match (LPM) route and the /32 unicast route. The Hash table is used for the /32 unicast entries. Each table has approximately 8000 routes.



Note

The Cisco Nexus 3064PQ offers half the scalability listed.



Note

IPv6 will use up to two entries for every route in the hardware.

Virtualization Support

The Unicast RIB and FIB support Virtual Routing and Forwarding instances (VRFs). For more information, see Chapter 14, “Configuring Layer 3 Virtualization.”

Licensing Requirements for the Unicast RIB and FIB

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	The unicast RIB and FIB 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. For a complete explanation of the Cisco NX-OS licensing scheme, see the <i>Cisco NX-OS Licensing Guide</i> .

Configuring the Unicast RIB and FIB

This section includes the following topics:

- [Displaying Module FIB Information, page 15-3](#)
- [Configuring Load Sharing in the Unicast FIB, page 15-4](#)
- [Displaying Routing and Adjacency Information, page 15-7](#)
- [Displaying Routing and Adjacency Information, page 15-7](#)
- [Triggering the Layer 3 Consistency Checker, page 15-8](#)
- [Clearing Forwarding Information in the FIB, page 15-9](#)
- [Estimating Memory Requirements for Routes, page 15-9](#)
- [Clearing Routes in the Unicast RIB, page 15-9](#)

**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.

Displaying Module FIB Information

You can display the FIB information on a switch.

DETAILED STEPS

To display the FIB information on a switch, use the following commands in any mode:

Command	Purpose
show ip fib adjacency Example: switch# show ip fib adjacency	Displays the adjacency information for FIB.
show forwarding ipv4 adjacency Example: switch# show forwarding ipv4 adjacency	Displays the adjacency information for IPv4.
show ip fib interfaces Example: switch# show ip fib interfaces	Displays the FIB interface information for IPv4.
show ip fib route Example: switch# show ip fib route	Displays the route table for IPv4.
show forwarding ipv4 route Example: switch# show forwarding ipv4 route	Displays the route table for IPv4.

This example shows how to display the FIB contents on a switch:

```
switch# show ip fib route
```

```
IPv4 routes for table default/base
```

```

-----+-----+-----
Prefix      | Next-hop      | Interface
-----+-----+-----
0.0.0.0/32   | Drop          | Null0
255.255.255.255/32 | Receive      | sup-eth1

```

Configuring Load Sharing in the Unicast FIB

Dynamic routing protocols, such as Open Shortest Path First (OSPF), support load balancing with equal-cost multipath (ECMP). The routing protocol determines its best routes based on the metrics configured for the protocol and installs up to the protocol-configured maximum paths in the unicast RIB. The unicast RIB compares the administrative distances of all routing protocol paths in the RIB and selects a best path set from all of the path sets installed by the routing protocols. The unicast RIB installs this best path set into the FIB for use by the forwarding plane.

The forwarding plane uses a load-sharing algorithm to select one of the installed paths in the FIB to use for a given data packet.

You can globally configure the following load-sharing settings:

- **load-share mode**—Selects the best path based on the destination address and port or the source and the destination address and port.
- **Universal ID**—Sets the random seed for the hash algorithm. You do not need to configure the Universal ID. Cisco NX-OS chooses the Universal ID if you do not configure it.

**Note**

Load sharing uses the same path for all packets in a given flow. A flow is defined by the load-sharing method that you configure. For example, if you configure source-destination load sharing, then all packets with the same source IP address and destination IP address pair follow the same path.

To configure the unicast FIB load-sharing algorithm, use the following command in global configuration mode:

Command	Purpose
<pre>ip load-sharing address {source destination port destination source-destination [port source-destination]} [universal-id seed]</pre> <p>Example: switch(config)# ip load-sharing address source-destination</p>	<p>Configures the unicast FIB load-sharing algorithm for data traffic. The <i>universal-id</i> range is from 1 to 4294967295.</p> <p>A new load sharing option is supported for ECMP, based on the source IP address. The new CLI option looks for the source IP address only on the traffic to load the balance, ignoring the rest of the parameters, for example, the destination IP address and the Layer 4 source port/destination port.</p>

Configuring Hash Offset

To avoid ECMP polarization in a multi-tier ECMP session, you must configure a different ECMP hash-offset on each tier. Starting with Release 6.0(2)U5(1), a new CLI for ECMP hash concatenation is introduced to achieve uniform distribution of the traffic across 16 way ECMP paths. The updated CLI support exists for Cisco Nexus 3100 Series switches and not on Cisco Nexus 3000 Series switches. You can configure the hash offset in the range of <0-15> in non-concatenate mode and in the range of <0-63> in concatenate mode.

In concatenate mode, if the hash-offset is set to 0 and concatenation is set, the **show running-config** command displays **hardware ecmp hash-offset 0 concatenation**. The hash-offset is programmed as per value. On downgrade, if concatenation is configured, the CAP check asks to remove the configuration.

In non-concatenate mode, if the hash-offset is set to 0 and concatenation is reset, the **show running-config** command does not display **hardware ecmp hash-offset 0**. The hash-offset is programmed as per value if the hash-offset value is in range 0-15. The hash-offset displays CLI error if the value is in range 16-63 (The non-concatenated mode supports hash-offset for 0-15 range).

The hash-offset in **show running-config** is visible as per the configured value.

To configure an ECMP hash-offset, use the following commands in global configuration mode:

Command	Purpose
#hardware ecmp hash-offset ?	Configures the ECMP hash-offset. The range is from 0 to 63. The hash offset in the range <0-15> is for the non-concatenate mode. The hash offset in the range <0-63> is for the concatenate mode.

Command	Purpose
#hardware ecmp hash-offset <i>number</i> Example: switch(config)# hardware ecmp hash-offset 5	Configures the ECMP hash-offset in the non-concatenate mode. The range is from 0 to 15. The default value is 0.
#hardware ecmp hash-offset <0-63> concatenation Example: switch(config)# hardware ecmp hash-offset 63 concatenation	Configures the ECMP hash-offset in the concatenate mode. Note Concatenation support exists for Cisco Nexus 3100 series switches and not for Cisco Nexus 3000 series switches. This CLI generates an error on Cisco Nexus 3000 series switches.

You can use the ECMP hash-offset configured by using the **hardware ecmp hash-offset** command along with different universal IDs configured by using the **ip load-sharing address** command to produce various hash results in the load-sharing algorithm.

Configuring Hash Polynomial

Starting with Release 6.0(2)U5(1), new CLI is added for the CRC configuration.

Command	Purpose
switch# config t	Enters configuration mode.
switch(config)# hardware ecmp ? hash-offset Configure hash offset hash-polynomial Configure hash polynomial	Displays hash-offset and hash-polynomial as the configuration options for hardware ECMP.
switch config)# hardware ecmp hash-polynomial ? CRC16 Hash polynomial CRC16 CRC32HI Hash polynomial CRC32 HI	Displays CRC16 and CRC32HI as configuration options for Hash polynomial.
switch config)# show running-config	Displays the running configuration.

To display the unicast FIB load-sharing algorithm, use the following command in any mode:

Command	Purpose
show ip load-sharing	Displays the unicast FIB load-sharing algorithm for data traffic.
Example: switch(config)# show ip load-sharing	

To display the route that the unicast RIB and FIB use for a particular source address and destination address, use the following command in any mode:

Command	Purpose
show routing hash <i>source-addr</i> <i>dest-addr</i> [ip-proto <i>ip-protocol</i>] [<i>source-l4-port</i> <i>dest-l4-port</i>] [vrf <i>vrf-name</i>] Example: switch# show routing hash 1.1.1.6.5.5 5.3 ip-proto 0x11 10 234	Displays the route that the unicast RIB FIB use for a source and destination address pair. The source address and destination address format is x.x.x.x. The source port and destination port range is from 1 to 65535. The VRF name can be any case-sensitive, alphanumeric string up to 64 characters. The ip-proto option corresponds to the protocol field of the IP header.

This example shows how to display the route selected for a source/destination pair:

```
switch# show routing hash 1.1.1.6.5.5.3 ip-proto 0x11 10 234
Load-share parameters used for software forwarding:
load-share mode: address source-destination port source-destination
Universal-id seed: 0xe05e2e85
Invoking pc_ic_ecmp_resolution
Hash for VRF "default"
Hashing to path *Eth1/29%
For route:
5.5.5.0/24 ubest/mbest: 3/0
    *via 2.2.2.1, Eth1/18, [1/0], 00:14:14, static
    *via 3.3.3.1, Eth1/29, [1/0], 00:14:14, static
    *via 4.4.4.1, Eth1/34, [1/0], 00:14:14, static
```

Displaying Routing and Adjacency Information

You can display the routing and adjacency information.

To display the routing and adjacency information, use the following commands in any mode:

Command	Purpose
show ip route [<i>route-type</i> interface <i>int-type number</i> next-hop] Example: switch# show ip route	Displays the unicast route table. The <i>route-type</i> argument can be a single route prefix, direct, static, or a dynamic route protocol. Use the ? keyword to see the supported interfaces.
show ip adjacency [<i>prefix</i> <i>interface-type number</i> [summary] non-best] [detail] [vrf <i>vrf-id</i>] Example: switch# show ip adjacency	Displays the adjacency table. The argument ranges are as follows: <ul style="list-style-type: none"> <i>prefix</i>—Any IPv4 prefix address. <i>interface-type number</i>—Use the ? keyword to see the supported interfaces. <i>vrf-id</i>—Any case-sensitive, alphanumeric string up to 32 characters.
show ip routing [<i>route-type</i> interface <i>int-type number</i> next-hop recursive-next-hop summary updated { since until } <i>time</i>] Example: switch# show routing summary	Displays the unicast route table. The <i>route-type</i> argument can be a single route prefix, direct, static, or a dynamic route protocol. Use the ? keyword to see the supported interfaces.

This example shows how to display the unicast route table:

```
switch# show ip route
IP Route Table for VRF "default"
'' denotes best ucast next-hop
''' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]

192.168.0.2/24, ubest/mbest: 1/0, attached
    *via 192.168.0.32, Eth1/5, [0/0], 22:34:09, direct
192.168.0.32/32, ubest/mbest: 1/0, attached
    *via 192.168.0.32, Eth1/5, [0/0], 22:34:09, local
```

This example shows how to display the adjacency information:

```
switch# show ip adjacency

IP Adjacency Table for VRF default
Total number of entries: 2
Address      Age      MAC Address  Pref Source  Interface  Best
10.1.1.1     02:20:54  00e0.b06a.71eb 50  arp      mgmt0      Yes
10.1.1.253   00:06:27  0014.5e0b.81d1 50  arp      mgmt0      Yes
```

Triggering the Layer 3 Consistency Checker

You can manually trigger the Layer 3 consistency checker.

To manually trigger the Layer 3 consistency checker for IPv4 or IPv6 routes, use the following commands in global configuration mode:

Command	Purpose
test [ipv4] [ipv6] [unicast] forwarding inconsistency [vrf vrf-name] [module {1 all}]	Starts a Layer 3 consistency check. The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 32 characters. The slot to be checked can be either 1 or all.
Example: switch(config)# test forwarding inconsistency	

To stop the Layer 3 consistency checker for IPv4 or IPv6 routes, use the following commands in global configuration mode:

Command	Purpose
test forwarding [ipv4] [ipv6] [unicast] inconsistency [vrf vrf-name] [module {1 all}] stop	Stops a Layer 3 consistency check. The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 64 characters. The slot can be either 1 or all.
Example: switch(config)# test forwarding inconsistency stop	

To display the Layer 3 inconsistencies for IPv4 or IPv6 routes, use the following commands in any mode:

Command	Purpose
<pre>show forwarding [ipv4] [ipv6] inconsistency [vrf vrf-name] [module {1 all}]</pre> <p>Example:</p> <pre>switch(config)# show forwarding inconsistency</pre>	Displays the results of a Layer 3 consistency check. The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 32 characters. The slot can be either 1 or all.

Clearing Forwarding Information in the FIB

You can clear one or more entries in the FIB. Clearing a FIB entry does not affect the unicast RIB.



Caution

The **clear forwarding** command disrupts forwarding on the switch.

To clear an entry in the FIB, including a Layer 3 inconsistency, use the following command in any mode:

Estimating Memory Requirements for Routes

Command	Purpose
<pre>clear forwarding {ipv4} route {* prefix} [vrf vrf-name] [module {slot all}]</pre> <p>Example:</p> <pre>switch(config)# clear forwarding ipv4 route *</pre>	<p>Clears one or more entries from the FIB. The route options are as follows:</p> <ul style="list-style-type: none"> *—All routes. <i>prefix</i>—Any IP prefix. <p>The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 32 characters. The <i>slot</i> range is from 1 to 10.</p>

You can estimate the memory that a number of routes and next-hop addresses will use.

To estimate the memory requirements for routes, use the following command in any mode:

Command	Purpose
<pre>show routing memory estimate routes num-routes next-hops num-nexthops</pre> <p>Example:</p> <pre>switch# show routing memory estimate routes 1000 next-hops 1</pre>	Displays the memory requirements for routes. The <i>num-routes</i> range is from 1000 to 1000000. The <i>num-nexthops</i> range is from 1 to 16.

Clearing Routes in the Unicast RIB

You can clear one or more routes from the unicast RIB.



Caution

The ***** keyword is severely disruptive to routing.

To clear one or more entries in the unicast RIB, use the following commands in any mode:

Command	Purpose
<pre>clear ip route {<i>*</i> {<i>route</i> prefix/length}[next-hop interface]} [vrf vrf-name]</pre> <p>Example: switch(config)# clear ip route 10.2.2.2</p>	<p>Clears one or more routes from both the unicast RIB and all the module FIBs. The route options are as follows:</p> <ul style="list-style-type: none"> • <i>*</i>—All routes. • <i>route</i>—An individual IP route. • <i>prefix/length</i>—Any IP prefix. • <i>next-hop</i>—The next-hop address • <i>interface</i>—The interface to reach the next-hop address. <p>The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 32 characters.</p>
<pre>clear routing unicast [ip ipv4] {<i>*</i> {<i>route</i> <i>prefix/length</i>}[next-hop interface]} [vrf vrf-name]</pre> <p>Example: switch(config)# clear routing ip 10.2.2.2</p>	<p>Clears one or more routes from the unicast RIB. The route options are as follows:</p> <ul style="list-style-type: none"> • <i>*</i>—All routes. • <i>route</i>—An individual IP route. • <i>prefix/length</i>—Any IP prefix. • <i>next-hop</i>—The next-hop address • <i>interface</i>—The interface to reach the next-hop address. <p>The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 32 characters.</p>

Verifying the Unicast RIB and FIB Configuration

To display the unicast RIB and FIB configuration information, perform one of the following tasks:

Command	Purpose
show forwarding adjacency	Displays the adjacency table on a module.
show forwarding distribution { <i>clients</i> <i>fib-state</i> }	Displays the FIB distribution information.
show forwarding interfaces module slot	Displays the FIB information for a module.
show forwarding ipv4 route	Displays routes in the FIB.
show hardware forwarding dynamic-allocation status	Displays information about the TCAM allocation.
show ip adjacency	Displays the adjacency table.
show ip route	Displays IPv4 routes from the unicast RIB.
show routing	Displays routes from the unicast RIB.

Additional References

For additional information related to managing unicast RIB and FIB, see the following sections:

- Related Documents, page 15-11

Related Documents

Related Topic	Document Title
Unicast RIB and FIB CLI commands	<i>Cisco Nexus 3000 Series Command Reference</i>



Configuring Route Policy Manager

This chapter describes how to configure the Route Policy Manager on the Cisco NX-OS switch.

This chapter includes the following sections:

- [Information About Route Policy Manager, page 16-1](#)
- [Licensing Requirements for Route Policy Manager, page 16-5](#)
- [Guidelines and Limitations, page 16-5](#)
- [Default Settings, page 16-6](#)
- [Configuring Route Policy Manager, page 16-6](#)
- [Verifying the Route Policy Manager Configuration, page 16-17](#)
- [Configuration Examples for Route Policy Manager, page 16-17](#)
- [Related Topics, page 16-18](#)
- [Additional References, page 16-18](#)

Information About Route Policy Manager

Route Policy Manager supports route maps and IP prefix lists. These features are used for route redistribution. A prefix list contains one or more IPv4 network prefixes and the associated prefix length values. You can use a prefix list by itself in features such as Border Gateway Protocol (BGP) templates, route filtering, or redistribution of routes that are exchanged between routing domains.

Route maps can apply to both routes and IP packets. Route filtering and redistribution pass a route through a route map.

This section includes the following topics:

- [Prefix Lists, page 16-2](#)
- [MAC Lists, page 16-2](#)
- [Route Maps, page 16-3](#)
- [Route Redistribution and Route Maps, page 16-5](#)

Prefix Lists

You can use prefix lists to permit or deny an address or range of addresses. Filtering by a prefix list involves matching the prefixes of routes or packets with the prefixes listed in the prefix list. An implicit deny is assumed if a given prefix does not match any entries in a prefix list.

You can configure multiple entries in a prefix list and permit or deny the prefixes that match the entry. Each entry has an associated sequence number that you can configure. If you do not configure a sequence number, Cisco NX-OS assigns a sequence number automatically. Cisco NX-OS evaluates prefix lists starting with the lowest sequence number. Cisco NX-OS processes the first successful match for a given prefix. Once a match occurs, Cisco NX-OS processes the permit or deny statement and does not evaluate the rest of the prefix list.

**Note**

An empty prefix list permits all routes.

Prefix List Masks

Cisco NX-OS Release 6.0(2)U3(1) introduces masks for prefix lists. Masking uses the number 1 and the number 0 to specify how the software treats the corresponding IP address bits.

- A mask bit 0 means ignore the corresponding bit value.
- A mask bit 1 means check the corresponding bit value for an exact match.

You can use a prefix list to match the IP address in a route-map, which in turn is used in routing protocols during redistribution. The IP address is matched against the prefix list where the bits corresponding to the mask bit 1 are the same as the subnet provided in the prefix list.

By carefully setting masks, you can select a single, or several IP addresses for permit or deny tests.

The prefix list mask allows noncontiguous bits in the mask. You can thus define a range of even or odd numbered IP addresses.

MAC Lists

You can use MAC lists to permit or deny MAC address or range of addresses. A MAC list consists of a list of MAC addresses and optional MAC masks. A MAC mask is a wildcard mask that is logically AND-ed with the MAC address when the route map matches on the MAC list entry. Filtering by a MAC list involves matching the MAC address of packets with the MAC addresses listed in the MAC list. An implicit deny is assumed if a given MAC address does not match any entries in a MAC list.

You can configure multiple entries in a MAC list and permit or deny the MAC addresses that match the entry. Each entry has an associated sequence number that you can configure. If you do not configure a sequence number, Cisco NX-OS assigns a sequence number automatically. Cisco NX-OS evaluates MAC lists starting with the lowest sequence number. Cisco NX-OS processes the first successful match for a given MAC address. Once a match occurs, Cisco NX-OS processes the permit or deny statement and does not evaluate the rest of the MAC list.

Route Maps

You can use route maps for route redistribution. Route map entries consist of a list of match and set criteria. The match criteria specify match conditions for incoming routes or packets, and the set criteria specify the action taken if the match criteria are met.

You can configure multiple entries in the same route map. These entries contain the same route map name and are differentiated by a sequence number.

You create a route map with one or more route map entries arranged by the sequence number under a unique route map name. The route map entry has the following parameters:

- Sequence number
- Permission—permit or deny
- Match criteria
- Set changes

By default, a route map processes routes or IP packets in a linear fashion, that is, starting from the lowest sequence number. You can configure the route map to process in a different order using the **continue** statement, which allows you to determine which route map entry to process next.

Match Criteria

You can use a variety of criteria to match a route or IP packet in a route map. Some criteria, such as BGP community lists, are applicable only to a specific routing protocol, while other criteria, such as the IP source or the destination address, can be used for any route or IP packet.

When Cisco NX-OS processes a route or packet through a route map, it compares the route or packet to each of the match statements configured. If the route or packet matches the configured criteria, Cisco NX-OS processes it based on the permit or deny configuration for that match entry in the route map and any set criteria configured.

The match categories and parameters are as follows:

- BGP parameters—Match based on AS numbers, AS-path, community attributes, or extended community attributes.
- Prefix lists—Match based on an address or range of addresses.
- Multicast parameters—Match based on a rendezvous point, groups, or sources.
- Other parameters—Match based on IP next-hop address or packet length.

Set Changes

Once a route or packet matches an entry in a route map, the route or packet can be changed based on one or more configured set statements.

The set changes are as follows:

- BGP parameters—Change the AS-path, tag, community, extended community, dampening, local preference, origin, or weight attributes.
- Metrics—Change the route-metric, the route-tag, or the route-type.
- Other parameters—Change the forwarding address or the IP next-hop address.

Access Lists

IP access lists can match the packet to a number of IP packet fields such as the following:

- Source or destination IPv4 address
- Protocol
- Precedence
- ToS

AS Numbers for BGP

You can configure a list of AS numbers to match against BGP peers. If a BGP peer matches an AS number in the list and matches the other BGP peer configuration, BGP creates a session. If the BGP peer does not match an AS number in the list, BGP ignores the peer. You can configure the AS numbers as a list, a range of AS numbers, or you can use an AS-path list to compare the AS numbers against a regular expression.

AS-Path Lists for BGP

You can configure an AS-path list to filter inbound or outbound BGP route updates. If the route update contains an AS-path attribute that matches an entry in the AS-path list, the router processes the route based on the permit or deny condition configured. You can configure AS-path lists within a route map.

You can configure multiple AS-path entries in an AS-path list by using the same AS-path list name. The router processes the first entry that matches.

Community Lists for BGP

You can filter BGP route updates based on the BGP community attribute by using community lists in a route map. You can match the community attribute based on a community list, and you can set the community attribute using a route map.

A community list contains one or more community attributes. If you configure more than one community attribute in the same community list entry, then the BGP route must match all community attributes listed to be considered a match.

You can also configure multiple community attributes as individual entries in the community list by using the same community list name. In this case, the router processes the first community attribute that matches the BGP route, using the permit or deny configuration for that entry.

You can configure community attributes in the community list in one of the following formats:

- A named community attribute, such as **internet** or **no-export**.
- In *aa:nn* format, where the first two bytes represent the two-byte AS number and the last two bytes represent a user-defined network number.
- A regular expression.

See the *Cisco Nexus 3000 Series Command Reference* for more information on regular expressions.

Extended Community Lists for BGP

Extended community lists support 4-byte AS numbers. You can configure community attributes in the extended community list in one of the following formats:

- In *aa4:nn* format, where the first four bytes represent the four-byte AS number and the last two bytes represent a user-defined network number.
- A regular expression.

See the *Cisco Nexus 3000 Series Command Reference* for more information on regular expressions.

Cisco NX-OS supports generic-specific extended community lists, which provide similar functionality to regular community lists for four-byte AS numbers. You can configure generic-specific extended community lists with the following properties:

- Transitive—BGP propagates the community attributes across autonomous systems.
- Nontransitive—BGP removes community attributes before propagating the route to another autonomous system.

Route Redistribution and Route Maps

You can use route maps to control the redistribution of routes between routing domains. Route maps match on the attributes of the routes to redistribute only those routes that pass the match criteria. The route map can also modify the route attributes during this redistribution using the set changes.

The router matches redistributed routes against each route map entry. If there are multiple match statements, the route must pass all of the match criteria. If a route passes the match criteria defined in a route map entry, the actions defined in the entry are executed. If the route does not match the criteria, the router compares the route against subsequent route map entries. Route processing continues until a match is made or the route is processed by all entries in the route map with no match. If the router processes the route against all entries in a route map with no match, the router accepts the route (inbound route maps) or forwards the route (outbound route maps).

Licensing Requirements for Route Policy Manager

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	Route Policy Manager requires 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. For a complete explanation of the Cisco NX-OS licensing scheme, see the <i>Cisco NX-OS Licensing Guide</i> .

Guidelines and Limitations

Route Policy Manager has the following configuration guidelines and limitations:

- An empty route map denies all the routes.
- An empty prefix list permits all the routes.
- Without any match statement in a route-map entry, the permission (permit or deny) of the route-map entry decides the result for all the routes or packets.
- If referred policies (for example, prefix lists) within a match statement of a route-map entry return either a no-match or a deny-match, Cisco NX-OS fails the match statement and processes the next route-map entry.

- When you change a route map, Cisco NX-OS holds all the changes until you exit from the route-map configuration submode. Cisco NX-OS then sends all the changes to the protocol clients to take effect.
- Because you can use a route map before you define it, verify that all your route maps exist when you finish a configuration change.
- You can view the route-map usage for redistribution and filtering. Each individual routing protocol provides a way to display these statistics.

Default Settings

Table 16-1 lists the default settings for Route Policy Manager.

Table 16-1 *Default Route Policy Manager Parameters*

Parameters	Default
Route Policy Manager	Enabled

Configuring Route Policy Manager

Route Policy Manager configuration includes the following topics:

- [Configuring IP Prefix Lists, page 16-6](#)
- [Configuring MAC Lists, page 16-8](#)
- [Configuring AS-Path Lists, page 16-8](#)
- [Configuring Community Lists, page 16-9](#)
- [Configuring Extended Community Lists, page 16-11](#)
- [Configuring Route Maps, page 16-12](#)



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.

Configuring IP Prefix Lists

IP prefix lists match the IP packet or route against a list of prefixes and prefix lengths. You can create an IP prefix list for IPv4.

You can configure the prefix list entry to match the prefix length exactly, or to match any prefix with a length that matches the configured range of prefix lengths.

Use the **ge** and **lt** keywords to create a range of possible prefix lengths. The incoming packet or route matches the prefix list if the prefix matches and if the prefix length is greater than or equal to the **ge** keyword value (if configured) and less than or equal to the **lt** keyword value (if configured).

Use the **mask** keyword to define a range of possible contiguous or non-contiguous routes to be compared to the prefix address.

SUMMARY STEPS

1. **configure terminal**
2. (Optional) **ip prefix-list name description string**
3. **ip prefix-list name [seq number] [{permit | deny} prefix {[eq prefix-length] | [ge prefix-length] [le prefix-length]]} [mask mask]**
4. (Optional) **show ip prefix-list name**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	ip prefix-list name description string Example: switch(config)# ip prefix-list AllowPrefix description allows engineering server	(Optional) Adds an information string about the prefix list.
Step 3	ip prefix-list name [seq number] [{permit deny} prefix {[eq prefix-length] [ge prefix-length] [le prefix-length]]} [mask mask] Example: switch(config)# ip prefix-list AllowPrefix seq 10 permit 192.0.2.0 eq 24 switch(config)# ip prefix-list even permit 0.0.0.0/32 mask 0.0.0.1	Creates an IPv4 prefix list or adds a prefix to an existing prefix list. The prefix length is matched as follows: <ul style="list-style-type: none"> • eq—Matches the exact <i>prefix length</i>. • ge—Matches a prefix length that is equal to or greater than the configured <i>prefix length</i>. • le—Matches a prefix length that is equal to or less than the configured <i>prefix length</i>. • mask—Specifies the bits of a prefix address in a prefix list that are compared to the bits of the prefix address used in routing protocols during redistribution.
Step 4	show ip prefix-list name Example: switch(config)# show ip prefix-list AllowPrefix	(Optional) Displays information about prefix lists.
Step 5	copy running-config startup-config Example: switch# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to create an IPv4 prefix list with two entries and apply the prefix list to a BGP neighbor:

```
switch# configure terminal
switch(config)# ip prefix-list allowprefix seq 10 permit 192.0.2.0/24 eq 25
```

```
switch(config)# ip prefix-list allowprefix seq 20 permit 209.165.201.0/26 eq 27
switch(config)# router bgp 65536:20
switch(config-router)# neighbor 192.0.2.1/16 remote-as 65535:20
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# prefix-list allowprefix in
```

This example shows how to create an IPv4 prefix list with a match mask for odd IP addresses:

```
switch# configure terminal
switch(config)# ip prefix-list odd permit 0.0.0.1/32 mask 0.0.0.1
```

Configuring MAC Lists

You can configure a MAC list to permit or deny a range of MAC addresses.

SUMMARY STEPS

1. **configure terminal**
2. **mac-list name [seq number] {permit | deny} mac-address [mac-mask]**
3. (Optional) **show mac-list name**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	mac-list name [seq number] {permit deny} mac-address {mac-mask} Example: switch(config)# mac-list AllowMac seq 1 permit 0022.5579.a4c1 ffff.ffff.0000	Creates a MAC list or adds a MAC address to an existing MAC list. The <i>seq</i> range is from 1 to 4294967294. The <i>mac-mask</i> specifies the portion of the MAC address to match against and is in MAC address format.
Step 3	show mac-list name Example: switch(config)# show mac-list AllowMac	(Optional) Displays information about MAC lists.
Step 4	copy running-config startup-config Example: switch# copy running-config startup-config	(Optional) Saves this configuration change.

Configuring AS-Path Lists

You can specify an AS-path list filter on both inbound and outbound BGP routes. Each filter is an access list based on regular expressions. If the regular expression matches the representation of the AS-path attribute of the route as an ASCII string, then the permit or deny condition applies.

SUMMARY STEPS

1. **configure terminal**
2. **ip as-path access-list** *name* {**deny** | **permit**} *expression*
3. (Optional) **show ip as-path list** *name*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	ip as-path access-list <i>name</i> { deny permit } <i>expression</i> Example: switch(config)# ip as-path access-list Allow40 permit 40	Creates a BGP AS-path list using a regular expression.
Step 3	show ip as-path-access-list <i>name</i> Example: switch(config)# show ip as-path-access-list Allow40	(Optional) Displays information about as-path access lists.
Step 4	copy running-config startup-config Example: switch# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to create an AS-path list with two entries and apply the AS path list to a BGP neighbor:

```
switch# configure terminal
switch(config)# ip as-path access-list AllowAS permit 64510
switch(config)# ip as-path access-list AllowAS permit 64496
switch(config)# copy running-config startup-config
switch(config)# router bgp 65536:20
switch(config-router)# neighbor 192.0.2.1/16 remote-as 65535:20
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# filter-list AllowAS in
```

Configuring Community Lists

You can use community lists to filter BGP routes based on the community attribute. The community number consists of a 4-byte value in the *aa:nn* format. The first two bytes represent the autonomous system number, and the last two bytes represent a user-defined network number.

When you configure multiple values in the same community list statement, all community values must match to satisfy the community list filter. When you configure multiple values in separate community list statements, the first list that matches a condition is processed.

Use community lists in a match statement to filter BGP routes based on the community attribute.

SUMMARY STEPS

1. **configure terminal**
2. **ip community-list standard** *list-name* {deny | permit} [*community-list*] [*internet*] [*local-AS*] [*no-advertise*] [*no-export*]
or
ip community-list expanded *list-name* {deny | permit} *expression*
3. (Optional) **show ip community-list** *name*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	ip community-list standard <i>list-name</i> {deny permit} [<i>community-list</i>] [<i>internet</i>] [<i>local-AS</i>] [<i>no-advertise</i>] [<i>no-export</i>] Example: switch(config)# ip community-list standard BGPCommunity permit no-advertise 65536:20	Creates a standard BGP community list. The <i>list-name</i> can be any case-sensitive, alphanumeric string up to 63 characters. The <i>community-list</i> can be one or more communities in the <i>aa:nn</i> format.
	ip community-list expanded <i>list-name</i> {deny permit} <i>expression</i> Example: switch(config)# ip community-list expanded BGPComplex deny 50000:[0-9][0-9]_	Creates an expanded BGP community list using a regular expression.
Step 3	show ip community-list <i>name</i> Example: switch(config)# show ip community-list BGPCommunity	(Optional) Displays information about community lists.
Step 4	copy running-config startup-config Example: switch# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to create a community list with two entries:

```
switch# configure terminal
switch(config)# ip community-list standard BGPCommunity permit no-advertise 65536:20
switch(config)# ip community-list standard BGPCommunity permit local-AS no-export
```

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

Configuring Extended Community Lists

You can use extended community lists to filter BGP routes based on the community attribute. The community number consists of a 6-byte value in the *aa4:nn* format. The first four bytes represent the autonomous system number, and the last two bytes represent a user-defined network number.

When you configure multiple values in the same extended community list statement, all extended community values must match to satisfy the extended community list filter. When you configure multiple values in separate extended community list statements, the first list that matches a condition is processed.

Use extended community lists in a match statement to filter BGP routes based on the extended community attribute.

SUMMARY STEPS

1. **configure terminal**
2. **ip extcommunity-list standard** *list-name* {deny | permit} 4bytegeneric {transitive | non-transitive} *community1* [*community2*]
or
ip extcommunity-list expanded *list-name* {deny | permit} *expression*
3. (Optional) **show ip extcommunity-list** *name*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	ip extcommunity-list standard <i>list-name</i> {deny permit} 4bytegeneric {transitive nontransitive} <i>community1</i> [<i>community2</i> ...] Example: switch(config)# ip extcommunity-list standard BGPExtCommunity permit 4bytegeneric transitive 65536:20 ip extcommunity-list expanded <i>list-name</i> {deny permit} <i>expression</i> Example: switch(config)# ip extcommunity-list expanded BGPExtComplex deny 1.5:[0-9][0-9]_	Creates a standard BGP extended community list. The <i>community</i> can be one or more extended communities in the <i>aa4:nn</i> format. Creates an expanded BGP extended community list using a regular expression.

	Command	Purpose
Step 3	show ip community-list <i>name</i> Example: switch(config)# show ip community-list BGPCommunity	(Optional) Displays information about extended community lists.
Step 4	copy running-config startup-config Example: switch# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to create a generic-specific extended community list:

```
switch# configure terminal
switch(config)# ip extcommunity-list standard test1 permit 4bytegeneric transitive
65536:40 65536:60
switch(config)# copy running-config startup-config
```

Configuring Route Maps

You can use route maps for route redistribution or route filtering. Route maps can contain multiple match criteria and multiple set criteria.

Configuring a route map for BGP triggers an automatic soft clear or refresh of BGP neighbor sessions.

SUMMARY STEPS

1. **configure terminal**
2. **route-map** *map-name* [**permit** | **deny**] [*seq*]
3. (Optional) **continue** *seq*
4. (Optional) **exit**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	route-map <i>map-name</i> [permit deny] [<i>seq</i>] Example: switch(config)# route-map Testmap permit 10 switch(config-route-map)#	Creates a route map or enters route-map configuration mode for an existing route map. Use <i>seq</i> to order the entries in a route map.
Step 3	continue <i>seq</i> Example: switch(config-route-map)# continue 10	(Optional) Determines what sequence statement to process next in the route map. Used only for filtering and redistribution.

	Command	Purpose
Step 4	exit Example: switch(config-route-map)# exit	(Optional) Exits route-map configuration mode.
Step 5	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

You can configure the following optional match parameters for route maps in route-map configuration mode:



Note The **default-information originate** command ignores **match** statements in the optional route map.

Command	Purpose
match as-path <i>name</i> [<i>name...</i>] Example: switch(config-route-map)# match as-path Allow40	Matches against one or more AS-path lists. Create the AS-path list with the ip as-path access-list command.
match as-number { <i>number</i> [, <i>number...</i>] as-path-list <i>name</i> [<i>name...</i>] } Example: switch(config-route-map)# match as-number 33,50-60	Matches against one or more AS numbers or AS-path lists. Create the AS-path list with the ip as-path access-list command. The number range is from 1 to 65535. The AS-path list name can be any case-sensitive, alphanumeric string up to 63 characters.
match community <i>name</i> [<i>name...</i>] [exact-match] Example: switch(config-route-map)# match community BGPCommunity	Matches against one or more community lists. Create the community list with the ip community-list command.
match extcommunity <i>name</i> [<i>name...</i>] [exact-match] Example: switch(config-route-map)# match extcommunity BGPextCommunity	Matches against one or more extended community lists. Create the community list with the ip extcommunity-list command.
match interface <i>interface-type number</i> [<i>interface-type number...</i>] Example: switch(config-route-map)# match interface e 1/2	Matches any routes that have their next-hop out one of the configured interfaces. Use ? to find a list of supported interface types.
match ip address prefix-list <i>name</i> [<i>name...</i>] Example: switch(config-route-map)# match ip address prefix-list AllowPrefix	Matches against one or more IPv4 prefix lists. Use the ip prefix-list command to create the prefix list.

Command	Purpose
match ip next-hop prefix-list <i>name</i> <i>[name...]</i> Example: switch(config-route-map)# match ip next-hop prefix-list AllowPrefix	Matches the IPv4 next-hop address of a route to one or more IP prefix lists. Use the ip prefix-list command to create the prefix list.
match ip route-source prefix-list <i>name</i> <i>[name...]</i> Example: switch(config-route-map)# match ip route-source prefix-list AllowPrefix	Matches the IPv4 route source address of a route to one or more IP prefix lists. Use the ip prefix-list command to create the prefix list.
match mac-list <i>name</i> <i>[name...]</i> Example: switch(config-route-map)# match mac-list AllowMAC	Matches against one or more MAC lists. Use the mac-list command to create the MAC list.
match metric <i>value</i> [+- <i>deviation.</i>] <i>[value...]</i> Example: switch(config-route-map)# match mac-list AllowMAC	Matches the route metric against one or more metric values or value ranges. Use +- deviation argument to set a metric range. The route map matches any route metric that falls the range: <i>value - deviation</i> to <i>value + deviation</i> .
match route-type <i>route-type</i> Example: switch(config-route-map)# match route-type level 1 level 2	Matches against a type of route. The <i>route-type</i> can be one or more of the following: <ul style="list-style-type: none"> • external • internal • level-1 • level-2 • local • nssa-external • type-1 • type-2
match tag <i>tagid</i> <i>[tagid...]</i> Example: switch(config-route-map)# match tag 2	Matches a route against one or more tags for filtering or redistribution.
match vlan <i>vlan-id</i> <i>[vlan-range]</i> Example: switch(config-route-map)# match vlan 3, 5-10	Matches against a VLAN.

You can configure the following optional set parameters for route maps in route-map configuration mode:

Command	Purpose
set as-path {tag prepend {last-as number as-1 [as-2...]} Example: switch(config-route-map)# set as-path prepend 10 100 110	Modifies an AS-path attribute for a BGP route. You can prepend the configured <i>number</i> of last AS numbers or a string of particular AS-path values (<i>as-1 as-2...as-n</i>).
set comm-list name delete Example: switch(config-route-map)# set comm-list BGPCommunity delete	Removes communities from the community attribute of an inbound or outbound BGP route update. Use the ip community-list command to create the community list.
set community {none additive local-AS no-advertise no-export community-1 [community-2...]} Example: switch(config-route-map)# set community local-AS	Sets the community attribute for a BGP route update. Note When you use both the set community and set comm-list delete commands in the same sequence of a route map attribute, the deletion operation is performed before the set operation. Note Use the send-community command in BGP neighbor address-family configuration mode to propagate BGP community attributes to BGP peers.
set dampening halflife reuse suppress duration Example: switch(config-route-map)# set dampening 30 1500 10000 120	Sets the following BGP route dampening parameters: <ul style="list-style-type: none"> <i>halflife</i>—The range is from 1 to 45 minutes. The default is 15. <i>reuse</i>—The range is from is 1 to 20000 seconds. The default is 750. <i>suppress</i>—The range is from is 1 to 20000. The default is 2000. <i>duration</i>—The range is from is 1 to 255 minutes. The default is 60.
set extcomm-list name delete Example: switch(config-route-map)# set extcomm-list BGPExtCommunity delete	Removes communities from the extended community attribute of an inbound or outbound BGP route update. Use the ip extcommunity-list command to create the extended community list.

Command	Purpose
set extcommunity generic {transitive nontransitive} {none additive} community-1 [community-2...]} Example: switch(config-route-map)# set extcommunity generic transitive 1.0:30	Sets the extended community attribute for a BGP route update. Note When you use both the set extcommunity and set extcomm-list delete commands in the same sequence of a route map attribute, the deletion operation is performed before the set operation. Note Use the send-community command in BGP neighbor address-family configuration mode to propagate BGP extended community attributes to BGP peers.
set forwarding-address Example: switch(config-route-map)# set forwarding-address	Sets the forwarding address for OSPF.
set level {backbone level-1 level-1-2 level-2} Example: switch(config-route-map)# set level backbone	Sets what area to import routes to for IS-IS. The options for IS-IS are level-1, level-1-2, or level-2. The default is level-1.
set local-preference value Example: switch(config-route-map)# set local-preference 4000	Sets the BGP local preference value. The range is from 0 to 4294967295.
set metric [+ -]bandwidth-metric Example: switch(config-route-map)# set metric +100	Adds or subtracts from the existing metric value. The metric is in Kb/s. The range is from 0 to 4294967295.
set metric bandwidth [delay reliability load mtu] Example: switch(config-route-map)# set metric 33 44 100 200 1500	Sets the route metric values. Metrics are as follows: <ul style="list-style-type: none"> <i>metric0</i>—Bandwidth in Kb/s. The range is from 0 to 4294967295. <i>metric1</i>—Delay in 10-microsecond units. <i>metric2</i>—Reliability. The range is from 0 to 255 (100 percent reliable). <i>metric3</i>—Loading. The range is from 1 to 200 (100 percent loaded). <i>metric4</i>—MTU of the path. The range is from 1 to 4294967295.

Command	Purpose
set metric-type { external internal type-1 type-2 } Example: switch(config-route-map)# set metric-type internal	Sets the metric type for the destination routing protocol. The options are as follows: external—IS-IS external metric internal—IGP metric as the MED for BGP type-1—OSPF external type 1 metric type-2—OSPF external type 2 metric
set origin { egp <i>as-number</i> igp incomplete } Example: switch(config-route-map)# set origin incomplete	Sets the BGP origin attribute. The EGP <i>as-number</i> range is from 0 to 65535.
set tag <i>name</i> Example: switch(config-route-map)# set tag 33	Sets the tag value for the destination routing protocol. The <i>name</i> parameter is an unsigned integer.
set weight <i>count</i> Example: switch(config-route-map)# set weight 33	Sets the weight for the BGP route. The range is from 0 to 65535.

The **set metric-type internal** command affects an outgoing policy and an eBGP neighbor only. If you configure both the **metric** and **metric-type internal** commands in the same BGP peer outgoing policy, then Cisco NX-OS ignores the **metric-type internal** command.

Verifying the Route Policy Manager Configuration

To display the route policy manager configuration information, perform one of the following tasks:

Command	Purpose
show ip community-list [<i>name</i>]	Displays information about a community list.
show ip extcommunity-list [<i>name</i>]	Displays information about an extended community list.
show [ip] prefix-list [<i>name</i>]	Displays information about an IPv4 prefix list.
show route-map [<i>name</i>]	Displays information about a route map.

Configuration Examples for Route Policy Manager

This example shows how to use an address family to configure BGP so that any unicast routes from neighbor 209.0.2.1 are accepted if they match access list 1:

```
router bgp 64496
  address-family ipv4 unicast
    network 192.0.2.0/24
    network 209.165.201.0/27 route-map filterBGP
```

```

route-map filterBGP
  match ip next-hop prefix-list AllowPrefix
ip prefix-list AllowPrefix 10 permit 192.0.2.0 eq 24
ip prefix-list AllowPrefix 20 permit 209.165.201.0 eq 27

```

Related Topics

The following topics can give more information on Route Policy Manager:

- [Chapter 7, “Configuring Basic BGP”](#)

Additional References

For additional information related to implementing IP, see the following sections:

- [Related Documents, page 16-18](#)
- [Standards, page 16-18](#)

Related Documents

Related Topic	Document Title
Route Policy Manager CLI commands	<i>Cisco Nexus 3000 Series Command Reference</i>

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—



Configuring Policy-Based Routing

This chapter describes how to configure policy based routing on the Cisco NX-OS device.

This chapter includes the following sections:

- [Information About Policy-Based Routing, page 17-1](#)
- [Licensing Requirements for Policy-Based Routing, page 17-3](#)
- [Prerequisites for Policy-Based Routing, page 17-3](#)
- [Guidelines and Limitations for Policy-Based Routing, page 17-3](#)
- [Default Settings, page 17-4](#)
- [Configuring Policy-Based Routing, page 17-4](#)
- [Verifying the Policy-Based Routing Configuration, page 17-8](#)
- [Displaying Policy-Based Routing Statistics, page 17-8](#)
- [Clearing Policy-Based Routing Statistics, page 17-8](#)
- [Configuration Examples for Policy Based-Routing, page 17-8](#)
- [Related Topics, page 17-9](#)
- [Additional References, page 17-9](#)

Information About Policy-Based Routing

Policy-based routing allows you to configure a defined policy for IPv4 and IPv6 traffic flows, lessening reliance on routes derived from routing protocols. All packets received on an interface with policy-based routing enabled are passed through enhanced packet filters or route maps. The route maps dictate the policy, determining where to forward packets.

Route maps are composed of match and set statements that you can mark as permit or deny. You can interpret the statements as follows:

- If the packets match any route map statements, all the set statements are applied. One of these actions involves choosing the next-hop.
- If a statement is marked as deny, the packets that meet the match criteria are sent back through the normal forwarding channels and destination-based routing is performed.
- If the statement is marked as permit and the packets do not match any route-map statements, the packets are sent back through the normal forwarding channels and destination-based routing is performed.

For more information, see the [“Route Maps” section on page 16-3](#).

Policy-based routing includes the following features:

- Source-based routing—Routes traffic that originates from different sets of users through different connections across the policy routers.
- Load sharing—Distributes traffic among multiple paths based on the traffic characteristics.

This section includes the following topics:

- [Policy Route Maps, page 17-2](#)
- [Set Criteria for Policy-Based Routing, page 17-2](#)

Policy Route Maps

Route-Maps are used to filter routes that are distributed across various routing protocols and between different entities in a given routing protocol. Each entry in a route map contains a combination of match and set statements. The match statements define the criteria for whether appropriate packets meet the particular policy (that is, the conditions to be met). The set clauses explain how the packets should be routed once they have met the match criteria.

You can mark the route-map statements as permit or deny. If the statement is marked as a deny, the packets that meet the match criteria are sent back through the normal forwarding channels (destination-based routing is performed). If the statement is marked as permit and the packets meet the match criteria, all the set clauses are applied. If the statement is marked as permit and the packets do not meet the match criteria, those packets are also forwarded through the normal routing channel.



Note

Policy routing is specified on the interface that receives the packets, not on the interface from which the packets are sent.

Set Criteria for Policy-Based Routing

The set criteria in a route map is evaluated in the order listed in the route map. Set criteria specific to route maps used for policy-based routing are as follows:

1. List of specified IP addresses—The IP address can specify the adjacent next-hop router in the path toward the destination to which the packets should be forwarded. The first IP address associated with a currently up connected interface is used to route the packets.



Note

You can optionally configure the set criteria for next-hop addresses to load balance traffic across up to 16 IP addresses. In this case, Cisco NX-OS sends all traffic for each IP flow to a particular IP next-hop address.

2. List of default next-hop IP addresses—Route to the interface or the next-hop address specified by this set statement only if there is no explicit route for the destination address of the packet in the routing table.



Note

You can optionally configure the set criteria for the default next-hop addresses to load balance traffic across a maximum of 16 IP addresses. In this case, Cisco NX-OS sends all traffic for each IP flow to a particular IP next-hop address.

3. NULL interface—Traffic that matches the match statement is dropped if you use the set null interface.

If the packets do not meet any of the defined match criteria, those packets are routed through the normal destination-based routing process.

Licensing Requirements for Policy-Based Routing

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	Policy-based routing requires an Enterprise Services license. For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the <i>Cisco NX-OS Licensing Guide</i> .

Prerequisites for Policy-Based Routing

Policy-based routing has the following prerequisites:

- Install the correct license.
- You must enable policy-based routing (see the [“Enabling the Policy-Based Routing Feature” section on page 17-4](#)).
- Assign an IP address on the interface and bring the interface up before you apply a route map on the interface for policy-based routing.
- Because the default size of IPv4 and IPv6 TCAM regions is 0, you must carve the PBR TCAMs and reload the switch before applying PBR policy on an interface (see the *Cisco Nexus 3000 Series NX-OS Security Configuration Guide*.)

Guidelines and Limitations for Policy-Based Routing

Policy-based routing has the following configuration guidelines and limitations:

- A policy-based routing route map can have only one match or set statement per route-map statement.
- A **match** command cannot refer to more than one ACL in a route map used for policy-based routing.
- An ACL used in a policy-based routing route map cannot include a deny statement.
- The same route map can be shared among different interfaces for policy-based routing as long as the interfaces belong to the same virtual routing and forwarding (VRF) instance.
- For Cisco Nexus 3000 Series switches, the maximum size of the IPv4 PBR TCAM can be 1024 by carving from other regions. For Cisco Nexus 3100 Series switches, this maximum size can be 2816 by carving from other regions.
- For Cisco Nexus 3000 Series switches, the maximum size of the IPv6 PBR TCAM can be 512 by carving from other regions. For Cisco Nexus 3100 Series switches, this maximum size can be 512 by carving from other regions.

Default Settings

Table 17-1 lists the default settings for policy-based routing parameters.

Table 17-1 *Default Policy-based Routing Parameters*

Parameters	Default
Policy-based routing	Disabled

Configuring Policy-Based Routing

This section includes the following topics:

- [Enabling the Policy-Based Routing Feature, page 17-4](#)
- [Configuring a Route Policy, page 17-5](#)



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 the Policy-Based Routing Feature

You must enable the policy-based routing feature before you can configure a route policy.

SUMMARY STEPS

1. **configure terminal**
2. **feature pbr**
3. (Optional) **show feature**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	feature pbr Example: switch(config)# feature pbr	Enables the policy-based routing feature.

	Command	Purpose
Step 3	show feature Example: switch(config)# show feature	(Optional) Displays enabled and disabled features.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

Use the **no feature pbr** command to disable the policy-based routing feature and remove all associated configuration.

Command	Purpose
no feature pbr Example: switch(config)# no feature pbr	Disables policy-based routing and removes all associated configuration.

Configuring a Route Policy

You can use route maps in policy-based routing to assign routing policies to the inbound interface. See the [“Configuring Route Maps” section on page 16-12](#).

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. **ip policy route-map** *map-name*
or
ipv6 policy route-map *map-nam*
4. (Optional) **exit**
5. (Optional) **exit**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
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 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	ip policy route-map map-name Example: switch(config-if)# ip policy route-map Testmap	Assigns a route map for IPv4 policy-based routing to the interface.
	ipv6 policy route-map map-name Example: switch(config-if)# ipv6 policy route-map TestIPv6map	Assigns a route map for IPv6 policy-based routing to the interface.
Step 4	exit Example: switch(config-route-map)# exit	(Optional) Exits route-map configuration mode.
Step 5	exit Example: switch(config)# exit	(Optional) Exits global configuration mode.
Step 6	copy running-config startup-config Example: switch# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to add a route map to an interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ip policy route-map Testmap
switch(config-if)# exit
switch(config)# copy running-config startup-config
```

You can configure the following optional match parameters for route maps in route-map configuration mode:

Command	Purpose
match ip address access-list-name name [name...] Example: switch(config-route-map)# match ip address access-list-name ACL1	Matches an IPv4 address against one or more IP access control lists (ACLs). This command is used for policy-based routing and is ignored by route filtering or redistribution.
match ipv6 address access-list-name name [name...] Example: switch(config-route-map)# match ipv6 address access-list-name ACLv6	Matches an IPv6 address against one or more IPv6 ACLs. This command is used for policy-based routing and is ignored by route filtering or redistribution.

You can configure the following optional set parameters for route maps in route-map configuration mode:

Command	Purpose
set ip next-hop address1 [address2...] {load-share} Example: switch(config-route-map)# set ip next-hop 192.0.2.1	Sets the IPv4 next-hop address for policy-based routing. This command uses the first valid next-hop address if multiple addresses are configured. Use the optional load-share keyword to load balance traffic across a maximum of 16 next-hop addresses.
set ip default next-hop address1 [address2...] {load-share} Example: switch(config-route-map)# set ip default next-hop 192.0.2.2	Sets the IPv4 next-hop address for policy-based routing when there is no explicit route to a destination. This command uses the first valid next-hop address if multiple addresses are configured. Use the optional load-share keyword to load balance traffic across a maximum of 16 next-hop addresses.
set ipv6 next-hop address1 [address2...] {load-share} Example: switch(config-route-map)# set ipv6 next-hop 2001:0DB8::1	Sets the IPv6 next-hop address for policy-based routing. This command uses the first valid next-hop address if multiple addresses are configured. Use the optional load-share keyword to load balance traffic across a maximum of 16 next-hop addresses.
set ipv6 default next-hop address1 [address2...] Example: switch(config-route-map)# set ipv6 default next-hop 2001:0DB8::2	Sets the IPv6 next-hop address for policy-based routing when there is no explicit route to a destination. This command uses the first valid next-hop address if multiple addresses are configured.

Cisco NX-OS routes the packet as soon as it finds a next-hop and an interface.

Verifying the Policy-Based Routing Configuration

To display policy-based routing configuration information, perform one of the following tasks:

Command	Purpose
show [ip ipv6] policy [name]	Displays information about an IPv4 or IPv6 policy.
show route-map [name] pbr-statistics	Displays policy statistics.

Use the **route-map map-name pbr-statistics** to enable policy statistics. Use the **clear route-map map-name pbr-statistics** to clear these policy statistics.

Displaying Policy-Based Routing Statistics

Use the **show route-map rmap-name pbr-statistics** command to display the statistics for policy-based routing. The statistics are maintained for each route-map sequence. It shows the number of packets that are policy-routed based on the match condition in a given route-map sequence. All other packets that are routed using the default routing table (could be due to unreachable next-hops in the **set** command) are also displayed. The PBR statistics collection must be turned on before any statistics can be shown.

This example shows how to display PBR statistics:

```
switch(config)# show route-map pbr-sample pbr-statistics
```

Clearing Policy-Based Routing Statistics

Use the **clear route-map rmap-name pbr-statistics** command to clear the counters maintained for PBR statistics of a route-map.

This example shows how to clear PBR statistics:

```
switch(config)# clear route-map pbr-sample pbr-statistics
```

Configuration Examples for Policy Based-Routing

This example shows how to configure a simple route policy on an interface:

```
feature pbr
ip access-list pbr-sample
  permit tcp host 10.1.1.1 host 192.168.2.1 eq 80
!
route-map pbr-sample
  match ip address pbr-sample
  set ip next-hop 192.168.1.1
!
route-map pbr-sample pbr-statistics

interface ethernet 1/2
  ip policy route-map pbr-sample
```

The following output verifies this configuration:

```
n3000# show route-map pbr-sample

route-map pbr-sample, permit, sequence 10
  Match clauses:
    ip address (access-lists): pbr-sample
  Set clauses:
    ip next-hop 192.168.1.1

n3000# show route-map pbr-sample pbr-statistics

route-map pbr-sample, permit, sequence 10
  Policy routing matches: 84 packets
```

Related Topics

The following topics can give more information on Policy Based Routing:

- [Chapter 16, “Configuring Route Policy Manager”](#)

Additional References

For additional information related to implementing IP, see the following sections:

- [Related Documents, page 17-10](#)
- [Standards, page 17-10](#)

Related Documents

Related Topic	Document Title
Policy-based routing CLI commands	<i>Cisco Nexus 3000 Series NX-OS Unicast Routing Command Reference, Release 6.x</i>

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

Feature History for Policy-Based Routing

[Table 17-2](#) lists the release history for this feature.

Table 17-2 Feature History for Policy-Based Routing

Feature Name	Releases	Feature Information
Policy-based routing	6.0(2)U2(1)	This feature was introduced.



Configuring Bidirectional Forwarding Detection

This chapter describes how to configure Bidirectional Forwarding Detection (BFD).

This chapter includes the following sections:

- [Information About BFD, page 18-1](#)
- [Licensing Requirements for BFD, page 18-3](#)
- [Prerequisites for BFD, page 18-3](#)
- [Guidelines and Limitations, page 18-3](#)
- [Default Settings, page 18-4](#)
- [Configuring BFD, page 18-5](#)
- [Configuring BFD for IPv6, page 18-16](#)
- [Monitoring BFD, page 18-23](#)

Information 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.

This section includes the following topics:

- [Asynchronous Mode, page 18-1](#)
- [BFD Detection of Failures, page 18-2](#)
- [BFD Echo Function, page 18-2](#)
- [Security, page 18-2](#)
- [Virtualization Support, page 18-3](#)

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 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.

BFD Detection of Failures

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 liveness, 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.

When a failure occurs in the network, the following occurs:

1. The BFD neighbor session with the BFD neighbor router is torn down.
2. BFD notifies the local BFD process that the BFD neighbor is no longer reachable.
3. The local BFD process tears down the BFD neighbor relationship.
4. If an alternative path is available, the routers immediately start converging on it.

**Note**

The BFD failure detection occurs in less than a second.

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 asymmetrical 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.

Virtualization Support

BFD supports virtual routing and forwarding instances (VRFs).

Licensing Requirements for BFD

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	BFD requires 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. For a complete explanation of the Cisco NX-OS licensing scheme, see the <i>Cisco NX-OS Licensing Guide</i> .

Prerequisites for BFD

BFD has the following prerequisites:

- You must enable the BFD feature (see the [“Enabling the BFD Feature” section on page 18-6](#)).
- For any client protocols that you want to enable BFD on, you enable BFD in that client protocol. See the [“Configuring BFD for IPv6” section on page 18-16](#).
- For any client protocols that you want to enable BFD on, you enable BFD in that client protocol.
- Disable Internet Control Message Protocol (ICMP) redirect messages on a BFD-enabled interfaces.
- See other detailed prerequisites that are listed with the configuration tasks.
- Beginning with Cisco NX-OS Release 6.0(2)U4(1), BFD for IPv6 is supported.

Guidelines and Limitations

BFD has the following configuration guidelines and limitations:

- BFD supports BFD version 1.
- BFD supports IPv4, IPv6, OSPFv2, BGPv4 and static routes.
- BFD supports single-hop BFD.
- BFD for BGP supports single-hop eBGP and iBGP with an update source.
- BFD supports the following Layer 3 interfaces—physical interfaces, port channels, subinterfaces, and VLAN interfaces (SVI).
- BFD does not support authentication for BFD or per-link BFD sessions on a port channel.
- 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.
- Port channel configuration limitations:
 - For Layer 3 port channels used by BFD, you must enable the Link Aggregation Control Protocol (LACP) on the port channel.

- For Layer 2 port channels used by SVI sessions, you must enable LACP on the port channel.
- SVI limitations:
 - 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.

**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 reenabling 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.

- Cisco NX-OS does not distribute the BFD operation to compatible modules to offload the CPU for BFD packet processing.
- BFD does not support stateless restarts and in-service software upgrades (ISSUs).
- If you want to enable BFD for a peer reachable through a port channel, you must configure LACP on the port channel.
- BFD echo mode and Unicast Reverse Path Forwarding (URPF) are mutually exclusive and cannot both be enabled on a BFD interface. If you want to configure an interface for BFD, you must disable either BFD echo mode or URPF.
- HSRP for IPv6, VRRPv3 for IPv6 are not supported with BFD.
- Static IPv6 routes are supported with BFD.
- The BFD echo function is supported only when using global IPv6 addresses and not when using IPv6 link-local addresses.

Default Settings

Table 18-1 lists the default settings for BFD parameters.

Table 18-1 **Default BFD Parameters**

Parameters	Default
BFD feature	Disabled
Required minimum receive interval	250 milliseconds
Desired minimum transmit interval	250 milliseconds
Detect multiplier	3
Echo function	Enabled
Mode	Asynchronous
Port channel	Logical mode (one session per source-destination pair address)
Slow timer	2000 milliseconds

Configuring BFD

This section includes the following topics:

- [Configuration Hierarchy, page 18-5](#)
- [Task Flow for Configuring BFD, page 18-5](#)
- [Enabling the BFD Feature, page 18-6](#)
- [Configuring Global BFD Parameters, page 18-6](#)
- [Configuring BFD on an Interface, page 18-8](#)
- [Configuring BFD on a Port Channel, page 18-9](#)
- [Configuring the BFD Echo Function, page 18-10](#)
- [Configuring BFD on BGP, page 18-12](#)
- [Configuring BFD on PIM, page 18-13](#)
- [Configuring BFD on OSPFv2, page 18-14](#)
- [Configuring BFD for Static Routes, page 18-15](#)
- [Configuring Global BFD Parameters for IPv6, page 18-16](#)
- [Configuring Per Interface BFD Parameters for IPv6, page 18-17](#)
- [Configuring BFD for IPv6 on OSPFv3, page 18-18](#)
- [Configuring BFD on IPv6 Static Routes, page 18-19](#)
- [Configuring BFD Echo Mode for IPv6, page 18-20](#)
- [Configuring BFD Session Echo Interval for IPv6, page 18-21](#)
- [Configuring a BFD Echo Interface for IPv6, page 18-21](#)
- [Configuring BFD Slow Timer for IPv6, page 18-22](#)

Configuration Hierarchy

You can configure BFD at the global level, VRF level, at the interface or port channel level, or at the subinterface level (for physical interfaces and port channels). The VRF configuration overrides global configuration. The interface or port channel configuration overrides VRF or global configuration. On supported interfaces, the subinterface-level configuration overrides the interface or port channel configuration. See the “[Configuring BFD on BGP](#)” section on [page 18-12](#) for more information.

For physical ports that are members of a port channel, the member port inherits the master port channel BFD configuration. The member port subinterfaces can override the master port channel BFD configuration.

Task Flow for Configuring BFD

Follow these steps to configure BFD:

-
- Step 1** [Enabling the BFD Feature.](#)

Step 2 [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.

SUMMARY STEPS

1. **configure terminal**
2. **feature bfd**
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	feature bfd Example: switch(config)# feature bfd	Enables the BFD feature.
Step 3	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

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 on page 18-8](#) to override these global session parameters on an interface.

BEFORE YOU BEGIN

Enable the BFD feature. See the [“Enabling the BFD Feature” section on page 18-6](#).

SUMMARY STEPS

1. **configure terminal**
2. **bfd interval *mintx* *min_rx* *msec* *multiplier* *value***

3. **bfd slow-timer** [*interval*]
4. (Optional) **show running-config bfd**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	bfd interval <i>mintx</i> min_rx <i>msec</i> multiplier <i>value</i> Example: switch(config)# bfd interval 250 min_rx 250 multiplier 3	Configures the BFD session parameters for all BFD sessions on the device. You can override these values by configuring the BFD session parameters on an interface. The <i>mintx</i> and <i>msec</i> range is from 250 to 999 milliseconds and the default is 250. The multiplier range is from 3 to 50. The multiplier default is 3. To return to the default settings, use the no bfd interval command.
Step 3	bfd slow-timer [<i>interval</i>] Example: switch(config)# bfd slow-timer 2000	Configures the slow timer. 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. The range is from 1000 to 30000 milliseconds. The default is 2000. To return to the default settings, use the no bfd slow-timer command.
Step 4	exit Example: switch(config)# exit switch#	Returns to EXEC mode.
Step 5	show running-config bfd Example: switch# show running-config bfd	(Optional) Displays the BFD running configuration.
Step 6	copy running-config startup-config Example: switch# copy running-config startup-config	(Optional) Saves this 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

Enable the BFD feature. See the [“Enabling the BFD Feature”](#) section on page 18-6.

SUMMARY STEPS

1. **configure terminal**
2. **interface *int-if***
3. **bfd interval *mintx* min_rx *msec* multiplier *value***
4. (Optional) **show running-config bfd**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>int-if</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode. Use the ? keyword to display the supported interfaces.
Step 3	bfd interval <i>mintx</i> min_rx <i>msec</i> multiplier <i>value</i> Example: switch(config-if)# bfd interval 250 min_rx 250 multiplier 3	Configures the BFD session parameters for all BFD sessions on the interface. This command overrides the global BFD session parameters. The <i>mintx</i> and <i>msec</i> range is from 250 to 999 milliseconds and the default is 250. The multiplier range is from 3 to 50. The multiplier default is 3. To return to the default settings, use the no bfd interval command.
Step 4	exit Example: switch(config-if)# exit switch (config)#	Exits interface configuration mode.
Step 5	exit Example: switch (config)# exit switch#	Exits configuration mode and returns to EXEC mode.

	Command	Purpose
Step 6	show running-config bfd Example: switch# show running-config bfd	(Optional) Displays the BFD running configuration.
Step 7	copy running-config startup-config Example: switch# copy running-config startup-config	(Optional) Saves this configuration change.

Configuring BFD on a Port Channel

You can configure the BFD session parameters for all BFD sessions on a port channel. For example, if the BFD session for one link on a port channel is up, BFD informs client protocols, such as BGP, 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, unless you configure subinterface-level BFD parameters on a member port. See the [“Configuring BFD on BGP” section on page 18-12](#) for more information.

BEFORE YOU BEGIN

Ensure that you enable LACP on the port channel before you enable BFD.

Enable the BFD feature. See the [“Enabling the BFD Feature” section on page 18-6](#).

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *number*
3. (Optional) **bfd interval** *mintx min_rx msec multiplier value*
4. (Optional) **show running-config bfd**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface port-channel <i>number</i> Example: switch(config)# interface port-channel 2 switch(config-if)#	Enters port channel configuration mode. Use the ? keyword to display the supported number range.

	Command	Purpose
Step 3	bfd interval <i>mintx</i> min_rx <i>msec</i> multiplier <i>value</i> Example: switch(config-if)# bfd interval 250 min_rx 250 multiplier 3	(Optional) Configures the BFD session parameters for all BFD sessions on the port channel. This command overrides the global BFD session parameters. The <i>mintx</i> and <i>msec</i> range is from 250 to 999 milliseconds and the default is 250. The multiplier range is from 3 to 50. The multiplier default is 3. To return to the default settings, use the no bfd interval command.
Step 4	exit Example: switch(config-if)# exit switch (config)#	Exits interface configuration mode.
Step 5	exit Example: switch (config)# exit switch#	Exits configuration mode, returns to EXEC mode.
Step 6	show running-config bfd Example: switch# show running-config bfd	(Optional) Displays the BFD running configuration.
Step 7	copy running-config startup-config Example: switch# copy running-config startup-config	(Optional) Saves this 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 on page 18-6.

Configure the BFD session parameters. See the [“Configuring Global BFD Parameters”](#) section on page 18-6 or the [“Configuring BFD on an Interface”](#) section on page 18-8.

Ensure that Internet Control Message Protocol (ICMP) redirect messages are disabled on BFD-enabled interfaces. Use the **no ip redirects** command on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **bfd slow-timer** *echo-interval*
3. **interface** *int-if*
4. **bfd echo**

5. (Optional) **show running-config bfd**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	bfd slow-timer <i>echo-interval</i> Example: switch(config)# bfd slow-timer 2000	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. To return to the default settings, use the no bfd slow-timer command.
Step 3	interface <i>int-if</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode. Use the ? keyword to display the supported interfaces.
Step 4	bfd echo Example: switch(config-if)# bfd echo	Enables the echo function. The default is enabled. To disable the echo function, use the no bfd echo command.
Step 5	exit Example: switch(config-if)# exit switch (config)#	Exits interface configuration mode.
Step 6	exit Example: switch (config)# exit switch#	Exits configuration mode and returns to EXEC mode.
Step 7	show running-config bfd Example: switch# show running-config bfd	(Optional) Displays the BFD running configuration.
Step 8	copy running-config startup-config Example: switch# copy running-config startup-config	(Optional) Saves this configuration change.

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 on page 18-6.
- Configure the BFD session parameters. See the [“Configuring Global BFD Parameters”](#) section on page 18-6 or the [“Configuring BFD on an Interface”](#) section on page 18-8.
- Enable the BGP feature. See the [“Enabling the BGP Feature”](#) section on page 7-11

SUMMARY STEPS

1. **configure terminal**
2. **router bgp *as-number***
3. **neighbor {*ip-address* | *ipv6-address*} remote-as *as-number***
4. **bfd**
5. (Optional) **show running-config bgp**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>as-number</i> Example: switch(config)# router bgp 64496 switch(config-router)#	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.
Step 3	neighbor {<i>ip-address</i> <i>ipv6-address</i>} remote-as <i>as-number</i> Example: switch(config-router)# neighbor 209.165.201.1 remote-as 64497 switch(config-router-neighbor)#	Configures the IPv4 or IPv6 address and AS number for a remote BGP peer. The <i>ip-address</i> format is x.x.x.x. The <i>ipv6-address</i> format is A:B::C:D.
Step 4	bfd Example: switch(config-router-neighbor)# bfd	Enables BFD for this BGP peer.

	Command	Purpose
Step 5	show running-config bgp Example: switch(config-router-neighbor)# show running-config bgp	(Optional) Displays the BGP running configuration.
Step 6	copy running-config startup-config Example: switch(config-router-neighbor)# copy running-config startup-config	(Optional) Saves this configuration change.

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 on page 18-6.

Enable the PIM feature. See the *Cisco Nexus 3000 Series NX-OS Multicast Routing Configuration Guide* for more information.

SUMMARY STEPS

1. **configure terminal**
2. **ip pim bfd**
3. **interface** *type slot/port*
4. (Optional) **ip pim bfd-instance** [**disable**]
5. (Optional) **show running-config pim**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	ip pim bfd Example: switch(config)# ip pim bfd	Enables BFD for PIM.
Step 3	interface <i>type slot/port</i> Example: switch(config)# interface ethernet 2/1 switch(config-if)#	Enters interface configuration mode. Use the ? keyword to display the supported interfaces.

	Command	Purpose
Step 4	ip pim bfd-instance [disable] Example: switch(config-if)# ip pim bfd-instance	(Optional) Enables or disables BFD on a PIM interface. The default is disabled.
Step 5	show running-config pim Example: switch(config)# show running-config pim	(Optional) Displays the PIM running configuration.
Step 6	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

Configuring BFD on OSPFv2

You can configure BFD for the Open Shortest Path First Protocol (OSPFv2).

BEFORE YOU BEGIN

Enable the BFD feature. See the [“Enabling the BFD Feature”](#) section on page 18-6.

Configure the BFD session parameters. See the [“Configuring Global BFD Parameters”](#) section on page 18-6 or the [“Configuring BFD on an Interface”](#) section on page 18-8.

Enable the OSPFv2 feature. See the [“Enabling the OSPFv2 Feature”](#) section on page 4-13.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf process-id**
3. **bfd**
4. (Optional) **show running-config ospf**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf process-id Example: switch(config)# router ospf 64496 switch(config-router)#	Creates a new OSPFv2 process with a configured id.

	Command	Purpose
Step 3	bfd Example: switch(config-router)# bfd	Enables BFD for this OSPFv2 peer. The default value is disabled.
Step 4	interface int-if Example: switch(config-router)# interface ethernet 1/2 switch (config-if)#	Enters interface configuration mode. Use the ? keyword to display the supported interfaces.
Step 5	[no] ip ospf bfd disable Example: switch(config-if)# ip ospf bfd disable	(Optional) Disables BFD on a OSPFv2 interface. The default value is enabled.
Step 6	show running-config ospf Example: switch(config)# show running-config ospf	(Optional) Displays the OSPFv2 running configuration.
Step 7	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

Configuring BFD for 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 on page 18-6.

SUMMARY STEPS

1. **configure terminal**
2. (Optional) **vrf context** *vrf-name*
3. **ip route route interface if** { *nh-address* | *nh-prefix* }
4. **ip route static bfd interface** { *nh-address* | *nh-prefix* }
5. (Optional) **show ip route static** [*vrf vrf-name*]
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	vrf context <i>vrf-name</i> Example: switch(config)# vrf context Red switch(config-vrf)#	(Optional) Enters VRF configuration mode.
Step 3	ip route <i>route interface {nh-address nh-prefix}</i> Example: switch(config-vrf)# ip route 192.0.2.1 ethernet 1/2 192.0.2.4	Creates a static route. Use the ? keyword to display the supported interfaces.
Step 4	ip route static bfd <i>interface {nh-address nh-prefix}</i> Example: switch(config-vrf)# ip route static bfd ethernet 1/2 192.0.2.4	Enables BFD for all static routes on an interface. Use the ? keyword to display the supported interfaces.
Step 5	show ip route static [<i>vrf vrf-name</i>] Example: switch(config-vrf)# show ip route static vrf Red	(Optional) Displays the static routes.
Step 6	copy running-config startup-config Example: switch(config-vrf)# copy running-config startup-config	(Optional) Saves this configuration change.

Configuring BFD for IPv6

Configuring Global BFD Parameters for IPv6

BEFORE YOU BEGIN

You can specify either the IPv4 or the IPv6 address family when you configure BFD parameters.

SUMMARY STEPS

1. **configure terminal**
2. **bfd** [*ipv4 | ipv6*] **interval** [*interval min_rx milliseconds multiplier interval-multiplier*]

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	bfd [ipv4 ipv6] interval [interval min_rx milliseconds multiplier interval-multiplier] Example: switch(config)# bfd ipv6 interval 50 min_rx 50 multiplier 3	Configures the BFD session parameters for all BFD sessions in the specified address family on the device. The Tx and Rx intervals range between 50 and 999 milliseconds. The multiplier ranges between 3 and 50.

Configuring Per Interface BFD Parameters for IPv6

BEFORE YOU BEGIN

BFD must be enabled on the device.

SUMMARY STEPS

1. **configure terminal**
2. **interface type slot/port**
3. **bfd [ipv4 | ipv6] interval [interval min_rx milliseconds multiplier interval-multiplier]**

DETAILED STEPS

	Command	Purpose
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 1/2 switch(config-if)#	Enters interface configuration mode. Use the ? keyword to display the supported interfaces
Step 3	bfd [ipv4 ipv6] interval [interval min_rx milliseconds multiplier interval-multiplier] Example: switch(config-if)# bfd ipv6 interval 50 min_rx 50 multiplier 3	Configures the BFD session parameters for all BFD sessions in the specified address family on the device. The Tx and Rx intervals range between 50 and 999 milliseconds. The multiplier ranges between 3 and 50.

Configuring BFD for IPv6 on OSPFv3

You can configure BFD for IPv6 on the Open Shortest Path First Protocol (OSPFv3).

BEFORE YOU BEGIN

- Enable the BFD feature. See the [“Enabling the BFD Feature”](#) section on page 18-6.
- Enable the OSPFv3 feature. See the [“Enabling OSPFv3”](#) section on page 5-14
- Configure the BFD session parameters. See the [“Configuring Global BFD Parameters for IPv6”](#) section on page 18-16 or the [“Configuring Per Interface BFD Parameters for IPv6”](#) section on page 18-17.
- Enable the OSPFv3 feature. See the *Cisco Nexus 3000 Series NX-OS Unicast Routing Configuration Guide* for more information.

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 process-id**
3. **bfd**
4. (Optional) **show running-config ospf**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospfv3 <i>process-id</i> Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv2 process with a configured id.
Step 3	bfd Example: switch(config-router)# bfd	Enables BFD for this OSPFv3 peer. The default value is disabled.
Step 4	show running-config ospfv3 Example: switch(config-router)# show running-config ospfv3	(Optional) Displays the OSPFv3 running configuration.
Step 5	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

Configuring BFD on IPv6 Static Routes

You can configure BFD for all IPv6 static routes on an interface.

PREREQUISITES

- Ensure that BFD is enabled on the devices at each end of the static route.
- Ensure that the BFD session parameters are configured.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context *vrf-name***
3. **ipv6 route *route interface {nh-address | nh-prefix}***
4. **ipv6 route static bfd *network-interface {nh-address | nh-prefix}***
5. (Optional) **show bfd neighbors**
6. (Optional) **show ipv6 route static**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	vrf context <i>vrf-name</i> Example: switch(config)# vrf context Red switch(config-vrf)#	(Optional) Enters VRF configuration mode to configure BFD on an IPv6 static route. Specifies the VRF for the route to be tracked.
Step 3	ipv6 route <i>route interface {nh-address nh-prefix}</i> Example: switch(config-vrf)# ipv6 route 1::5/64 ethernet 1/3 2::2	Creates an IPv6 static route. <ul style="list-style-type: none">Specify the IPv6 address for the route argumentUse the ? keyword to display the supported interfaces.Specify the next-hop (nh) address or prefix for this static route.
Step 4	ipv6 route static bfd <i>network-interface {nh-address nh-prefix}</i> Example: switch(config-vrf)# ipv6 route static bfd ethernet 1/3 2::2	Enables BFD for all IPv6 static routes on an interface. <ul style="list-style-type: none">Use the ? keyword to display the supported interfaces.Specify the next-hop (nh) address or prefix for this static route.
Step 5	show bfd neighbors Example: switch(config-vrf)# show bfd neighbors	(Optional) Displays information about BFD neighbors.
Step 6	show ipv6 route static Example: switch(config-vrf)# show ipv6 route static vrf Red	(Optional) Displays the static routes.
Step 7	copy running-config startup-config Example: switch(config-vrf)# copy running-config startup-config	(Optional) Saves this configuration change.

Configuring BFD Echo Mode for IPv6

The echo function is enabled by default. You can disable it for IPv4, IPv6, or all address families.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. **[no] bfd [ipv4 | ipv6] echo**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode. Use the ? keyword to display the supported interfaces
Step 3	[no] bfd [ipv4 ipv6] echo Example: switch(config-if)# bfd ipv6 echo	Enables the echo function. The default is enabled. To disable the echo function for the specified address family, use the no form of the command.

Configuring BFD Session Echo Interval for IPv6

You can configure the echo Rx interval for BFD sessions.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. **[no] bfd [ipv4 | ipv6] echo-rx-interval** *interval*

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode. Use the ? keyword to display the supported interfaces
Step 3	[no] bfd [ipv4 ipv6] echo-rx-interval <i>interval</i> Example: switch(config-if)# bfd ipv6 echo-rx-interval 500	Configures the echo Rx interval for the BFD session. The interval can range between 50 and 999 milliseconds. To return the echo interval to the default value of 250 milliseconds, use the no form of the command.

Configuring a BFD Echo Interface for IPv6

Perform this task to configure the loopback interface as the source address for all echo frames.

SUMMARY STEPS

1. **configure terminal**
2. **interface loopback** *number*
3. **ip address** *ip-address mask*
4. **ipv6 address** {*ipv6-address / prefix-length* | *prefix-name sub-bits / prefix-length*}

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface loopback <i>number</i> Example: switch(config)# interface loopback 50 switch(config-if)#	Creates a loopback interface and enters interface configuration mode.
Step 3	ip address <i>ip-address mask</i> Example: switch(config-if)# ip address 192.108.1.27 255.255.255.0	Configures the IP address for the interface.
Step 4	ipv6 address { <i>ipv6-address/prefix-length</i> <i>prefix-name sub-bits / prefix-length</i> } Example: switch(config-if)# ipv6 address 2001:DB8:c18:1::3/64	Configures the IPv6 address as the source address for all echo frames.

Configuring BFD Slow Timer for IPv6

Echo mode is enabled by default. You can configure the slow-timer value and disable or enable echo mode for an address family.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. **bfd** [*ipv4* | *ipv6*] **slow-timer** [*interval*]

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode. Use the ? keyword to display the supported interfaces
Step 3	bfd [ipv4 ipv6] slow-timer [<i>interval</i>] Example: switch(config-if)# bfd ipv6 slow-timer 6000	Configures the slow timer, in milliseconds, used in the echo function for the specified address family.

Verifying the BFD Configuration

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

Command	Purpose
show running-config bfd	Displays the running BFD configuration.
show startup-config bfd	Displays the BFD configuration that will be applied on the next system startup.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 3000 Series NX-OS Interfaces Command Reference, Release 6.x*.

Monitoring BFD

Use the following commands to display BFD for IPv4 and IPv6 neighbors:

Command	Purpose
show bfd [ipv4 ipv6] neighbors [application <i>name</i>] [details]	Displays information about BFD for a supported application, such as BGP.
show bfd [ipv4 ipv6] neighbors [interface <i>int-if</i>] [details]	Displays information about BGP sessions on an interface.
show bfd [ipv4 ipv6] neighbors [dest-ip <i>ip-address</i>] [src-ip <i>ip-address</i>][details]	Displays information about the specified BGP session on an interface.
show bfd [ipv4 ipv6] neighbors [vrf <i>vrf-name</i>] [details]	Displays information about BFD for a VRF.

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 3000 Series Command Reference*.



PART 3

First-Hop Redundancy Protocols



Configuring HSRP

This chapter describes how to configure the Hot Standby Router Protocol (HSRP) on the Cisco NX-OS switch.

This chapter includes the following sections:

- [Information About HSRP, page 20-1](#)
- [Licensing Requirements for HSRP, page 20-6](#)
- [Prerequisites for HSRP, page 20-6](#)
- [Guidelines and Limitations, page 20-6](#)
- [Default Settings, page 20-6](#)
- [Configuring HSRP, page 20-7](#)
- [Verifying the HSRP Configuration, page 20-16](#)
- [Configuration Examples for HSRP, page 20-16](#)
- [Additional References, page 20-16](#)

Information About HSRP

HSRP is a first-hop redundancy protocol (FHRP) that allows a transparent failover of the first-hop IP router. HSRP provides first-hop routing redundancy for IP hosts on Ethernet networks configured with a default router IP address. You use HSRP in a group of routers for selecting an active router and a standby router. In a group of routers, the active router is the router that routes packets; the standby router is the router that takes over when the active router fails or when preset conditions are met.

Many host implementations do not support any dynamic router discovery mechanisms but can be configured with a default router. Running a dynamic router discovery mechanism on every host is not feasible for a number of reasons, including administrative overhead, processing overhead, and security issues. HSRP provides failover services to these hosts.

This section includes the following topics:

- [HSRP Overview, page 20-2](#)
- [HSRP for IPv4, page 20-3](#)
- [HSRP Versions, page 20-4](#)
- [HSRP Authentication, page 20-4](#)
- [HSRP Messages, page 20-4](#)

- [HSRP Load Sharing, page 20-4](#)
- [Object Tracking and HSRP, page 20-5](#)
- [Virtualization Support, page 20-5](#)

HSRP Overview

When you use HSRP, you configure the HSRP virtual IP address as the host's default router (instead of the IP address of the actual router). The virtual IP address is an IPv4 address that is shared among a group of routers that run HSRP.

When you configure HSRP on a network segment, you provide a virtual MAC address and a virtual IP address for the HSRP group. You configure the same virtual address on each HSRP-enabled interface in the group. You also configure a unique IP address and MAC address on each interface that acts as the real address. HSRP selects one of these interfaces to be the active router. The active router receives and routes packets destined for the virtual MAC address of the group.

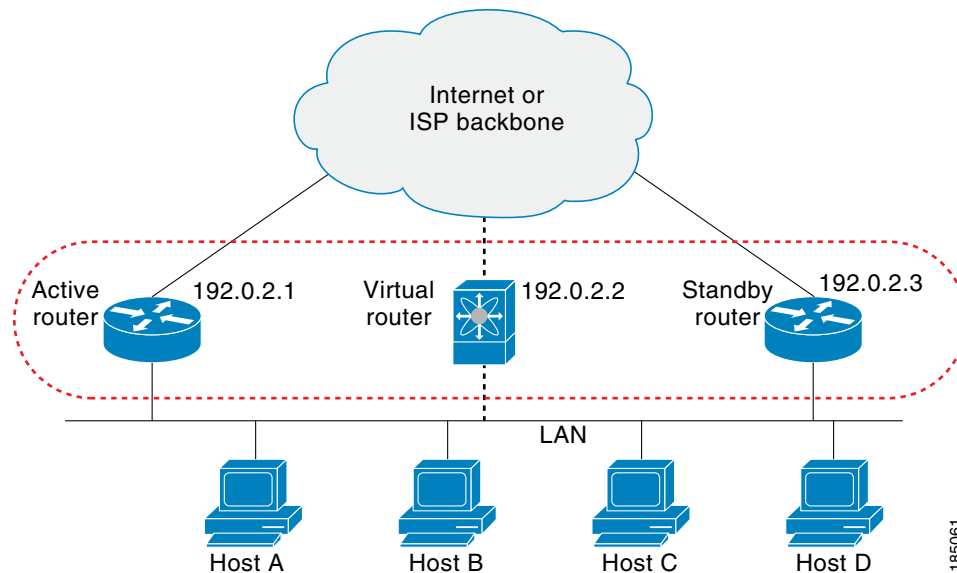
HSRP detects when the designated active router fails. At that point, a selected standby router assumes control of the virtual MAC and IP addresses of the HSRP group. HSRP also selects a new standby router at that time.

HSRP uses a priority mechanism to determine which HSRP-configured interface becomes the default active router. To configure an interface as the active router, you assign it with a priority that is higher than the priority of all the other HSRP-configured interfaces in the group. The default priority is 100, so if you configure just one interface with a higher priority, that interface becomes the default active router.

Interfaces that run HSRP send and receive multicast User Datagram Protocol (UDP)-based hello messages to detect a failure and to designate active and standby routers. When the active router fails to send a hello message within a configurable period of time, the standby router with the highest priority becomes the active router. The transition of packet forwarding functions between the active and standby router is completely transparent to all hosts on the network.

You can configure multiple HSRP groups on an interface.

[Figure 20-1](#) shows a network configured for HSRP. By sharing a virtual MAC address and a virtual IP address, two or more interfaces can act as a single virtual router.

Figure 20-1 HSRP Topology with Two Enabled Routers

The virtual router does not physically exist but represents the common default router for interfaces that are configured to provide backup to each other. You do not need to configure the hosts on the LAN with the IP address of the active router. Instead, you configure them with the IP address (virtual IP address) of the virtual router as their default router. If the active router fails to send a hello message within the configurable period of time, the standby router takes over, responds to the virtual addresses, and becomes the active router, assuming the active router duties. From the host perspective, the virtual router remains the same.

**Note**

Packets received on a routed port destined for the HSRP virtual IP address will terminate on the local router, regardless of whether that router is the active HSRP router or the standby HSRP router. This includes ping and Telnet traffic. Packets received on a Layer 2 (VLAN) interface destined for the HSRP virtual IP address will terminate on the active router.

HSRP for IPv4

HSRP routers communicate with each other by exchanging HSRP hello packets. These packets are sent to the destination IP multicast address 224.0.0.2 (reserved multicast address used to communicate to all routers) on UDP port 1985. The active router sources hello packets from its configured IP address and the HSRP virtual MAC address while the standby router sources hellos from its configured IP address and the interface MAC address, which may or may not be the burned-in address (BIA). The BIA is the last six bytes of the MAC address that is assigned by the manufacturer of the network interface card (NIC).

Because hosts are configured with their default router as the HSRP virtual IP address, hosts must communicate with the MAC address associated with the HSRP virtual IP address. This MAC address is a virtual MAC address, 0000.0C07.ACxy, where xy is the HSRP group number in hexadecimal based on the respective interface. For example, HSRP group 1 uses the HSRP virtual MAC address of 0000.0C07.AC01. Hosts on the adjoining LAN segment use the normal Address Resolution Protocol (ARP) process to resolve the associated MAC addresses.

HSRP version 2 uses the new IP multicast address 224.0.0.102 to send hello packets instead of the multicast address of 224.0.0.2, which is used by version 1. HSRP version 2 permits an expanded group number range of 0 to 4095 and uses a new MAC address range of 0000.0C9F.F000 to 0000.0C9F.FFFF.

HSRP Versions

Cisco NX-OS supports HSRP version 1 by default. You can configure an interface to use HSRP version 2.

HSRP version 2 has the following enhancements to HSRP version 1:

- Expands the group number range. HSRP version 1 supports group numbers from 0 to 255. HSRP version 2 supports group numbers from 0 to 4095.
- Uses the MAC address range from 0000.0C9F.F000 to 0000.0C9F.FFFF for IPv4. HSRP version 1 uses the MAC address range 0000.0C07.AC00 to 0000.0C07.ACFE.
- Adds support for MD5 authentication.

When you change the HSRP version, Cisco NX-OS reinitializes the group because it now has a new virtual MAC address.

HSRP version 2 has a different packet format than HSRP version 1. The packet format uses a type-length-value (TLV) format. HSRP version 2 packets received by an HSRP version 1 router are ignored.

HSRP Authentication

HSRP message digest 5 (MD5) algorithm authentication protects against HSRP-spoofing software and uses the industry-standard MD5 algorithm for improved reliability and security. HSRP includes the IPv4 address in the authentication TLVs.

HSRP Messages

Routers that are configured with HSRP exchange the following three types of multicast messages:

- Hello—The hello message conveys the HSRP priority and state information of the router to other HSRP routers.
- Coup—When a standby router wants to assume the function of the active router, it sends a coup message.
- Resign—A router that is the active router sends this message when it is about to shut down or when a router that has a higher priority sends a hello or coup message.

HSRP Load Sharing

HSRP allows you to configure multiple groups on an interface. You can configure two overlapping IPv4 HSRP groups to load share traffic from the connected hosts while providing the default router redundancy expected from HSRP. [Figure 20-2](#) shows an example of a load-sharing HSRP IPv4 configuration.

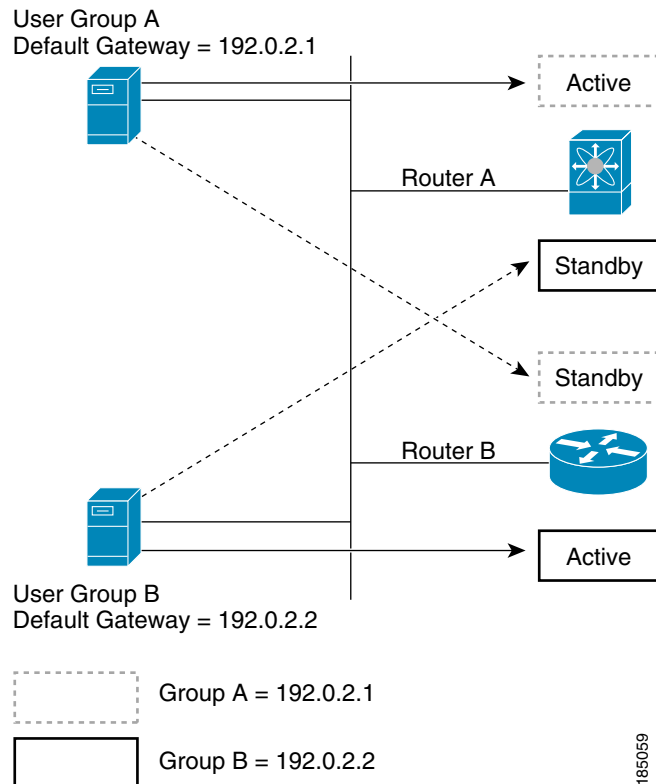
Figure 20-2 HSRP Load Sharing

Figure 20-2 shows two routers (A and B) and two HSRP groups. Router A is the active router for group A but is the standby router for group B. Similarly, router B is the active router for group B and the standby router for group A. If both routers remain active, HSRP load balances the traffic from the hosts across both routers. If either router fails, the remaining router continues to process traffic for both hosts.

Object Tracking and HSRP

You can use object tracking to modify the priority of an HSRP interface based on the operational state of another interface. Object tracking allows you to route to a standby router if the interface to the main network fails.

Two objects that you can track are the line protocol state of an interface or the reachability of an IP route. If the specified object goes down, Cisco NX-OS reduces the HSRP priority by the configured amount. For more information, see the [“Configuring HSRP Object Tracking” section on page 20-12](#).

Virtualization Support

HSRP supports Virtual Routing and Forwarding instances (VRFs). By default, Cisco NX-OS places you in the default VRF unless you specifically configure another VRF.

If you change the VRF membership of an interface, Cisco NX-OS removes all Layer 3 configuration, including HSRP.

For more information, see [Chapter 14, “Configuring Layer 3 Virtualization.”](#)

Licensing Requirements for HSRP

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	HSRP requires 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. For a complete explanation of the Cisco NX-OS licensing scheme, see the <i>Cisco NX-OS Licensing Guide</i> .
	Note Make sure the LAN Base Services license is installed on the switch to enable Layer 3 interfaces.

Prerequisites for HSRP

HSRP has the following prerequisites:

- You must enable the HSRP feature in a switch before you can configure and enable any HSRP groups.

Guidelines and Limitations

HSRP has the following configuration guidelines and limitations:

- The minimum hello timer value is 250 milliseconds.
- The minimum hold timer value is 750 milliseconds.
- You must configure an IP address for the interface that you configure HSRP on and enable that interface before HSRP becomes active.
- For IPv4, the virtual IP address must be in the same subnet as the interface IP address.
- We recommend that you do not configure more than one first-hop redundancy protocol on the same interface.
- HSRP version 2 does not interoperate with HSRP version 1. An interface cannot operate both version 1 and version 2 because both versions are mutually exclusive. However, the different versions can be run on different physical interfaces of the same router.
- You cannot change from version 2 to version 1 if you have configured groups above the group number range allowed for version 1 (0 to 255).
- Cisco NX-OS removes all Layer 3 configuration on an interface when you change the interface VRF membership, port channel membership, or when you change the port mode to Layer 2.

Default Settings

[Table 20-1](#) lists the default settings for HSRP parameters.

Table 20-1 **Default HSRP Parameters**

Parameters	Default
HSRP	Disabled
Authentication	Enabled as text for version 1, with cisco as the password
HSRP version	Version 1
Preemption	disabled
Priority	100
virtual MAC address	Derived from HSRP group number

Configuring HSRP

This section includes the following topics:

- [Enabling the HSRP Feature, page 20-7](#)
- [Configuring the HSRP Version, page 20-8](#)
- [Configuring an HSRP Group for IPv4, page 20-8](#)
- [Configuring the HSRP Virtual MAC Address, page 20-10](#)
- [Authenticating HSRP, page 20-10](#)
- [Configuring HSRP Object Tracking, page 20-12](#)
- [Configuring the HSRP Priority, page 20-14](#)
- [Customizing HSRP, page 20-14](#)

**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 the HSRP Feature

You must globally enable the HSRP feature before you can configure and enable any HSRP groups.

DETAILED STEPS

To enable the HSRP feature, use the following command in global configuration mode:

Command	Purpose
feature hsrp	Enables HSRP.
Example: switch(config)# feature hsrp	

To disable the HSRP feature and remove all associated configuration, use the following command in global configuration mode:

Command	Purpose
no feature hsrp Example: switch(config)# no feature hsrp	Disables HSRP.

Configuring the HSRP Version

You can configure the HSRP version. If you change the version for existing groups, Cisco NX-OS reinitializes HSRP for those groups because the virtual MAC address changes. The HSRP version applies to all groups on the interface.

To configure the HSRP version, use the following command in interface configuration mode:

Command	Purpose
hsrp version {1 2} Example: switch(config-if)# hsrp version 2	Configures the HSRP version. Version 1 is the default.

Configuring an HSRP Group for IPv4

You can configure an HSRP group on an IPv4 interface and configure the virtual IP address and virtual MAC address for the HSRP group.

BEFORE YOU BEGIN

Ensure that you have enabled the HSRP feature (see the [“Enabling the HSRP Feature”](#) section on page 20-7).

Cisco NX-OS enables an HSRP group once you configure the virtual IP address on any member interface in the group. You should configure HSRP attributes such as authentication, timers, and priority before you enable the HSRP group.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type number*
3. **no switchport**
4. **ip** *ip-address/length*
5. **hsrp group-number** [ipv4]
6. **ip** [*ip-address* [secondary]]
7. **exit**
8. **no shutdown**
9. (Optional) **show hsrp** [group *group-number*] [ipv4]
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface type number Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	ip ip-address/length Example: switch(config-if)# ip 192.0.2.2/8	Configures the IPv4 address of the interface.
Step 5	hsrp group-number [ipv4] Example: switch(config-if)# hsrp 2 switch(config-if-hsrp)#	Creates an HSRP group and enters hsrp configuration mode. The range for HSRP version 1 is from 0 to 255. The range is for HSRP version 2 is from 0 to 4095. The default value is 0.
Step 6	ip [ip-address [secondary]] Example: switch(config-if-hsrp)# ip 192.0.2.1	Configures the virtual IP address for the HSRP group and enables the group. This address should be in the same subnet as the IPv4 address of the interface.
Step 7	exit Example: switch(config-if-hsrp)# exit	Exits HSRP configuration mode.
Step 8	no shutdown Example: switch(config-if)# no shutdown	Enables the interface.
Step 9	show hsrp [group group-number] [ipv4] Example: switch(config-if)# show hsrp group 2	(Optional) Displays HSRP information.
Step 10	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

**Note**

You should use the **no shutdown** command to enable the interface after you finish the configuration.

This example shows how to configure an HSRP group on Ethernet 1/2:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
```

```

switch(config-if)# no switchport
switch(config-if)# ip 192.0.2.2/8
switch(config-if)# hsrp 2
switch(config-if-hsrp)# ip 192.0.2.1
switch(config-if-hsrp)# exit
switch(config-if)# no shutdown
switch(config-if)# copy running-config startup-config

```

Configuring the HSRP Virtual MAC Address

You can override the default virtual MAC address that HSRP derives from the configured group number.

To manually configure the virtual MAC address for an HSRP group, use the following command in hsrp configuration mode:

Command	Purpose
mac-address <i>string</i> Example: switch(config-if-hsrp)# mac-address 5000.1000.1060	Configures the virtual MAC address for an HSRP group. The string uses the standard MAC address format (xxxx.xxxx.xxxx).

To configure HSRP to use the burned-in MAC address of the interface for the virtual MAC address, use the following command in interface configuration mode:

Command	Purpose
hsrp use-bia [<i>scope interface</i>] Example: switch(config-if)# hsrp use-bia	Configures HSRP to use the burned-in MAC address of the interface for the HSRP virtual MAC address. You can optionally configure HSRP to use the burned-in MAC address for all groups on this interface by using the scope interface keywords.

Authenticating HSRP

You can configure HSRP to authenticate the protocol using cleartext or MD5 digest authentication. MD5 authentication uses a keychain.

BEFORE YOU BEGIN

Ensure that you have enabled the HSRP feature (see the [“Enabling the HSRP Feature”](#) section on page 20-7).

You must configure the same authentication and keys on all members of the HSRP group.

Ensure that you have created the keychain if you are using MD5 authentication.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*

3. **no switchport**
4. **hsrp group-number [ipv4]**
5. **authentication text string**
or
authentication md5 {keychain keychain | key-string {0 | 7} text [timeout seconds]}
6. (Optional) **show hsrp [group group-number]**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface interface-type slot/port Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	hsrp group-number [ipv4] Example: switch(config-if)# hsrp 2 switch(config-if-hsrp)#	Creates an HSRP group and enters HSRP configuration mode.
Step 5	authentication text string Example: switch(config-if-hsrp)# authentication text mypassword authentication md5 {keychain keychain key-string {0 7} text [timeout seconds]} Example: switch(config-if-hsrp)# authentication md5 keychain hsrp-keys	Configures cleartext authentication for HSRP on this interface. Configures MD5 authentication for HSRP on this interface. You can use a keychain or key string. If you use a key string, you can optionally set the timeout for when HSRP will only accept a new key. The range is from 0 to 32767 seconds.
Step 6	show hsrp [group group-number] Example: switch(config-if-hsrp)# show hsrp group 2	(Optional) Displays HSRP information.
Step 7	copy running-config startup-config Example: switch(config-if-hsrp)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure MD5 authentication for HSRP on Ethernet 1/2 after creating the keychain:

```
switch# configure terminal
switch(config)# key chain hsrp-keys
switch(config-keychain)# key 0
switch(config-keychain-key)# key-string 7 zqdest
switch(config-keychain-key) accept-lifetime 00:00:00 Jun 01 2008 23:59:59 Sep 12 2008
switch(config-keychain-key) send-lifetime 00:00:00 Jun 01 2008 23:59:59 Aug 12 2008
switch(config-keychain-key) key 1
switch(config-keychain-key) key-string 7 uaeqdyito
switch(config-keychain-key) accept-lifetime 00:00:00 Aug 12 2008 23:59:59 Dec 12 2008
switch(config-keychain-key) send-lifetime 00:00:00 Sep 12 2008 23:59:59 Nov 12 2008
switch(config-keychain-key)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# hsrp 2
switch(config-if-hsrp)# authenticate md5 keychain hsrp-keys
switch(config-if-hsrp)# copy running-config startup-config
```

Configuring HSRP Object Tracking

You can configure an HSRP group to adjust its priority based on the availability of other interfaces or routes. The priority of a switch can change dynamically if it has been configured for object tracking and the object that is being tracked goes down. The tracking process periodically polls the tracked objects and notes any value change. The value change triggers HSRP to recalculate the priority. The HSRP interface with the higher priority becomes the active router if you configure the HSRP interface for preemption.

HSRP supports tracked objects and track lists. See [Chapter 22, “Configuring Object Tracking”](#) for more information on track lists.

BEFORE YOU BEGIN

Ensure that you have enabled the HSRP feature (see the [“Enabling the HSRP Feature”](#) section on [page 20-7](#)).

SUMMARY STEPS

1. **configure terminal**
2. **track object-id interface interface-type number {ip routing | line-protocol}**
or
track object-id ip route ip-prefix/length reachability
3. **interface interface-type slot/port**
4. **no switchport**
5. **hsrp group-number [ipv4]**
6. **priority [value]**
7. **track object-number [decrement value]**
8. **preempt [delay [minimum seconds] [reload seconds] [sync seconds]]**
9. (Optional) **show hsrp interface interface-type number**
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	track object-id interface <i>interface-type number {ip routing line-protocol}</i> Example: switch(config)# track 1 interface ethernet 2/2 line-protocol switch(config-track)#	Configures the interface that this HSRP interface tracks. Changes in the state of the interface affect the priority of this HSRP interface as follows: <ul style="list-style-type: none"> You configure the interface and corresponding object number that you use with the track command in hsrp configuration mode. The line-protocol keyword tracks whether the interface is up. The ip keyword also checks that IP routing is enabled on the interface and an IP address is configured.
	track object-id ip route <i>ip-prefix/length reachability</i> Example: switch(config)# track 2 ip route 192.0.2.0/8 reachability switch(config-track)#	Creates a tracked object for a route and enters tracking configuration mode. The <i>object-id</i> range is from 1 to 500.
Step 3	interface interface-type slot/port Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 4	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 5	hsrp group-number [ipv4] Example: switch(config-if)# hsrp 2 switch(config-if-hsrp)#	Creates an HSRP group and enters hsrp configuration mode.
Step 6	priority [value] Example: switch(config-if-hsrp)# priority 254	Sets the priority level used to select the active router in an HSRP group. The range is from 0 to 255. The default is 100.
Step 7	track object-number [decrement value] Example: switch(config-if-hsrp)# track 1 decrement 20	Specifies an object to be tracked that affects the weighting of an HSRP interface. The <i>value</i> argument specifies a reduction in the priority of an HSRP interface when a tracked object fails. The range is from 1 to 255. The default is 10.

	Command	Purpose
Step 8	<pre>preempt [delay [minimum seconds] [reload seconds] [sync seconds]]</pre> <p>Example:</p> <pre>switch(config-if-hsrp)# preempt delay minimum 60</pre>	Configures the router to take over as the active router for an HSRP group if it has a higher priority than the current active router. This command is disabled by default. The range is from 0 to 3600 seconds.
Step 9	<pre>show hsrp interface interface-type number</pre> <p>Example:</p> <pre>switch(config-if-hsrp)# show hsrp interface ethernet 1/2</pre>	(Optional) Displays HSRP information for an interface.
Step 10	<pre>copy running-config startup-config</pre> <p>Example:</p> <pre>switch(config-if-hsrp)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

This example shows how to configure HSRP object tracking on Ethernet 1/2:

```
switch# configure terminal
switch(config)# track 1 interface ethernet 2/2 line-protocol
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# hsrp 2
switch(config-if-hsrp)# track 1 decrement 20
switch(config-if-hsrp)# copy running-config startup-config
```

Configuring the HSRP Priority

You can configure the HSRP priority on an interface. HSRP uses the priority to determine which HSRP group member acts as the active router.

To configure the HSRP priority, use the following command in interface configuration mode:

Command	Purpose
<pre>priority level [forwarding-threshold lower lower-value upper upper-value]</pre> <p>Example:</p> <pre>switch(config-if-hsrp)# priority 60 forwarding-threshold lower 40 upper 50</pre>	Sets the priority level used to select the active router in an HSRP group. The <i>level</i> range is from 0 to 255. The default is 100.

Customizing HSRP

You can optionally customize the behavior of HSRP. Be aware that as soon as you enable an HSRP group by configuring a virtual IP address, that group is now operational. If you first enable an HSRP group before customizing HSRP, the router could take control over the group and become the active router before you finish customizing the feature. If you plan to customize HSRP, you should do so before you enable the HSRP group.

To customize HSRP, use the following commands in hsrp configuration mode:

Command	Purpose
name <i>string</i> Example: switch(config-if-hsrp)# name HSRP-1	Specifies the IP redundancy name for an HSRP group. The <i>string</i> is from 1 to 255 characters. The default string has the following format: <i>hsrp-interface-short-name-group-id</i> . For example, <i>hsrp-Eth2/1-1</i> .
preempt [delay [minimum <i>seconds</i>] [reload <i>seconds</i>] [sync <i>seconds</i>]] Example: switch(config-if-hsrp)# preempt delay minimum 60	Configures the router to take over as an active router for an HSRP group if it has a higher priority than the current active router. This command is disabled by default. The range is from 0 to 3600 seconds.
timers [msec] <i>hellotime</i> [msec] <i>holdtime</i> Example: switch(config-if-hsrp)# timers 5 18	Configures the hello and hold time for this HSRP member as follows: <ul style="list-style-type: none"> <i>hellotime</i>—The interval between successive hello packets sent. The range is from 1 to 254 seconds. <i>holdtime</i>—The interval before the information in the hello packet is considered invalid. The range is from 3 to 255. <p>The optional msec keyword specifies that the argument is expressed in milliseconds, instead of the default seconds. The timer ranges for milliseconds are as follows:</p> <ul style="list-style-type: none"> <i>hellotime</i>—The interval between successive hello packets sent. The range is from 255 to 999 milliseconds. <i>holdtime</i>—The interval before the information in the hello packet is considered invalid. The range is from 750 to 3000 milliseconds.

To customize HSRP, use the following commands in interface configuration mode:

Command or Action	Purpose
hsrp delay minimum <i>seconds</i> Example: switch(config-if)# hsrp delay minimum 30	Specifies the minimum amount of time that HSRP waits after a group is enabled before participating in the group. The range is from 0 to 10000 seconds. The default is 0.
hsrp delay reload <i>seconds</i> Example: switch(config-if)# hsrp delay reload 30	Specifies the minimum amount of time that HSRP waits after reload before participating in the group. The range is from 0 to 10000 seconds. The default is 0.

Verifying the HSRP Configuration

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

Command	Purpose
show hsrp [group <i>group-number</i>]	Displays the HSRP status for all groups or one group.
show hsrp delay [interface <i>interface-type slot/port</i>]	Displays the HSRP delay value for all interfaces or one interface.
show hsrp [interface <i>interface-type slot/port</i>]	Displays the HSRP status for an interface.
show hsrp [group <i>group-number</i>] [interface <i>interface-type slot/port</i>] [active] [all] [init] [learn] [listen] [speak] [standby]	Displays the HSRP status for a group or interface for virtual forwarders in the active, init, learn, listen, or standby state. Use the all keyword to see all states, including disabled.
show hsrp [group <i>group-number</i>] [interface <i>interface-type slot/port</i>] [active] [all] [init] [learn] [listen] [speak] [standby] brief	Displays a brief summary of the HSRP status for a group or interface for virtual forwarders in the active, init, learn, listen, or standby state. Use the all keyword to see all states, including disabled.

Configuration Examples for HSRP

This example shows how to enable HSRP on an interface with MD5 authentication and interface tracking:

```
key chain hsrp-keys
key 0
  key-string 7 zqdest
  accept-lifetime 00:00:00 Jun 01 2008 23:59:59 Sep 12 2008
  send-lifetime 00:00:00 Jun 01 2008 23:59:59 Aug 12 2008
key 1
  key-string 7 uaeqdyito
  accept-lifetime 00:00:00 Aug 12 2008 23:59:59 Dec 12 2008
  send-lifetime 00:00:00 Sep 12 2008 23:59:59 Nov 12 2008
feature hsrp
track 2 interface ethernet 2/2 ip
interface ethernet 1/2
  no switchport
  ip address 192.0.2.2/8
  hsrp 1
    authenticate md5 keychain hsrp-keys
    priority 90
    track 2 decrement 20
    ip-address 192.0.2.10
  no shutdown
```

Additional References

For additional information related to implementing HSRP, see the following sections:

- [Related Documents, page 20-17](#)

- [MIBs, page 20-17](#)

Related Documents

Related Topic	Document Title
Configuring the Virtual Router Redundancy Protocol	Chapter 21, “Configuring VRRP”
HSRP CLI commands	<i>Cisco Nexus 3000 Series Command Reference</i>

MIBs

MIBs	MIBs Link
CISCO-HSRP-MIB	To locate and download MIBs, go to the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml



Configuring VRRP

This chapter describes how to configure the Virtual Router Redundancy Protocol (VRRP) on a switch

This chapter includes the following sections:

- [Information About VRRP, page 21-1](#)
- [Licensing Requirements for VRRP, page 21-5](#)
- [Guidelines and Limitations, page 21-6](#)
- [Default Settings, page 21-6](#)
- [Configuring VRRP, page 21-7](#)
- [Configuring VRRPv3, page 21-17](#)
- [Verifying the VRRPv2 Configuration, page 21-21](#)
- [Verifying the VRRPv3 Configuration, page 21-22](#)
- [Displaying VRRP Statistics, page 21-22](#)
- [Configuration Examples for VRRPv2, page 21-22](#)
- [Configuration Example for VRRPv3, page 21-24](#)
- [Additional References, page 21-24](#)

Information About VRRP

VRRP allows for transparent failover at the first-hop IP router, by configuring a group of routers to share a virtual IP address. VRRP selects a master router in that group to handle all packets for the virtual IP address. The remaining routers are in standby and take over if that the master router fails.

This section includes the following topics:

- [VRRP Operation, page 21-2](#)
- [VRRP Benefits, page 21-3](#)
- [Multiple VRRP Groups, page 21-3](#)
- [VRRP Router Priority and Preemption, page 21-4](#)
- [VRRP Advertisements, page 21-5](#)
- [VRRP Authentication, page 21-5](#)
- [VRRPv3, page 21-5](#)

- [Virtualization Support, page 21-5](#)

VRRP Operation

A LAN client can determine which router should be the first hop to a particular remote destination by using a dynamic process or static configuration. Examples of dynamic router discovery are as follows:

- Proxy ARP—The client uses Address Resolution Protocol (ARP) to get the destination it wants to reach, and a router will respond to the ARP request with its own MAC address.
- Routing protocol—The client listens to dynamic routing protocol updates (for example, from Routing Information Protocol [RIP]) and forms its own routing table.
- ICMP Router Discovery Protocol (IRDP) client—The client runs an Internet Control Message Protocol (ICMP) router discovery client.

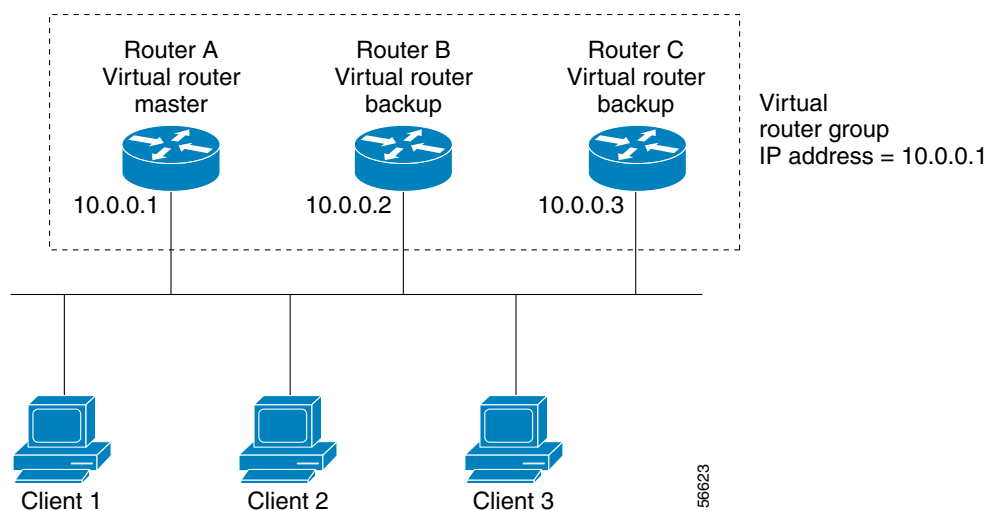
The disadvantage to dynamic discovery protocols is that they incur some configuration and processing overhead on the LAN client. Also, in the event of a router failure, the process of switching to another router can be slow.

An alternative to dynamic discovery protocols is to statically configure a default router on the client. Although, this approach simplifies client configuration and processing, it creates a single point of failure. If the default gateway fails, the LAN client is limited to communicating only on the local IP network segment and is cut off from the rest of the network.

VRRP can solve the static configuration problem by enabling a group of routers (a VRRP group) to share a single virtual IP address. You can then configure the LAN clients with the virtual IP address as their default gateway.

[Figure 21-1](#) shows a basic VLAN topology. In this example, Routers A, B, and C form a VRRP group. The IP address of the group is the same address that was configured for the Ethernet interface of Router A (10.0.0.1).

Figure 21-1 Basic VRRP Topology



Because the virtual IP address uses the IP address of the physical Ethernet interface of Router A, Router A is the master (also known as the IP address owner). As the master, Router A owns the virtual IP address of the VRRP group router and forwards packets sent to this IP address. Clients 1 through 3 are configured with the default gateway IP address of 10.0.0.1.

Routers B and C function as backups. If the master fails, the backup router with the highest priority becomes the master and takes over the virtual IP address to provide uninterrupted service for the LAN hosts. When router A recovers, it becomes the router master again. For more information, see the “[VRRP Router Priority and Preemption](#)” section.

**Note**

Packets received on a routed port destined for the VRRP virtual IP address will terminate on the local router, regardless of whether that router is the master VRRP router or a backup VRRP router. This includes ping and Telnet traffic. Packets received on a Layer 2 (VLAN) interface destined for the VRRP virtual IP address will terminate on the master router.

VRRP Benefits

The benefits of VRRP are as follows:

- **Redundance**—Enables you to configure multiple routers as the default gateway router, which reduces the possibility of a single point of failure in a network.
- **Load Sharing**—Allows traffic to and from LAN clients to be shared by multiple routers. The traffic load is shared more equitably among available routers.
- **Multiple VRRP groups**—Supports up to 255 VRRP groups on a router physical interface if the platform supports multiple MAC addresses. Multiple VRRP groups enable you to implement redundancy and load sharing in your LAN topology.
- **Multiple IP Addresses**—Allows you to manage multiple IP addresses, including secondary IP addresses. If you have multiple subnets configured on an Ethernet interface, you can configure VRRP on each subnet.
- **Preemption**—Enables you to preempt a backup router that has taken over for a failing master with a higher priority backup router that has become available.
- **Advertisement Protocol**—Uses a dedicated Internet Assigned Numbers Authority (IANA) standard multicast address (224.0.0.18) for VRRP advertisements. This addressing scheme minimizes the number of routers that must service the multicasts and allows test equipment to accurately identify VRRP packets on a segment. IANA has assigned the IP protocol number 112 to VRRP.
- The benefits of VRRPv3 are as follows:
 - Interoperability in multi-vendor environments.
 - Support for the IPv4 and IPv6 address families.

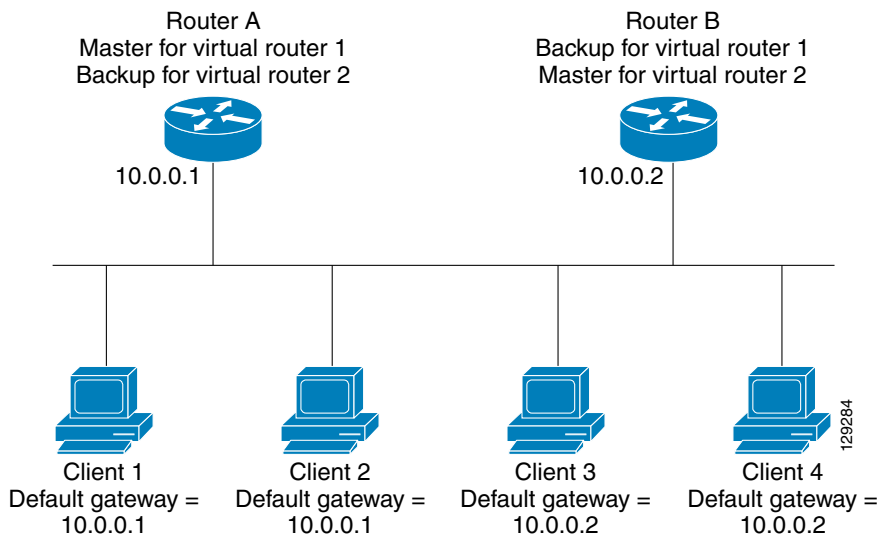
Multiple VRRP Groups

You can configure up to 255 VRRP groups on a physical interface. The actual number of VRRP groups that a router interface can support depends on the following factors:

- Router processing capability
- Router memory capability

In a topology where multiple VRRP groups are configured on a router interface, the interface can act as a master for one VRRP group and as a backup for one or more other VRRP groups.

[Figure 21-2](#) shows a LAN topology in which VRRP is configured so that Routers A and B share the traffic to and from clients 1 through 4. Routers A and B act as backups to each other if either router fails.

Figure 21-2 Load Sharing and Redundancy VRRP Topology

This topology contains two virtual IP addresses for two VRRP groups that overlap. For VRRP group 1, Router A is the owner of IP address 10.0.0.1 and is the master. Router B is the backup to router A. Clients 1 and 2 are configured with the default gateway IP address of 10.0.0.1.

For VRRP group 2, Router B is the owner of IP address 10.0.0.2 and is the master. Router A is the backup to router B. Clients 3 and 4 are configured with the default gateway IP address of 10.0.0.2.

VRRP Router Priority and Preemption

An important aspect of the VRRP redundancy scheme is the VRRP router priority because the priority determines the role that each VRRP router plays and what happens if the master router fails.

If a VRRP router owns the virtual IP address and the IP address of the physical interface, this router functions as the master. The priority of the master is 255.

Priority also determines if a VRRP router functions as a backup router and the order of ascendancy to becoming a master if the master fails.

When the master switch and the backup switches have the same priority value, the backup switch has a higher IP address, and initially the master switch is up. In this scenario, when the backup switch comes up, it should not disturb the state of master switch. Only when the master switch goes down, the backup switch becomes the master switch.

For example, if router A, the master in a LAN topology fails, VRRP must determine if backups B or C should take over. If you configure router B with priority 101 and router C with the default priority of 100, VRRP selects router B to become the master because it has the higher priority. If you configure routers B and C with the default priority of 100, VRRP selects the backup with the higher IP address to become the master.

VRRP uses preemption to determine what happens after a VRRP backup router becomes the master. With preemption enabled by default, VRRP will switch to a backup if that backup comes online with a priority higher than the new master. For example, if Router A is the master and fails, VRRP selects Router B (next in order of priority). If Router C comes online with a higher priority than Router B, VRRP selects Router C as the new master, even though Router B has not failed.

If you disable preemption, VRRP will only switch if the original master recovers or the new master fails.

VRRP Advertisements

The VRRP master sends VRRP advertisements to other VRRP routers in the same group. The advertisements communicate the priority and state of the master. Cisco NX-OS encapsulates the VRRP advertisements in IP packets and sends them to the IP multicast address assigned to the VRRP group. Cisco NX-OS sends the advertisements once every second by default, but you can configure a different advertisement interval.

VRRP Authentication

VRRP supports the following authentication mechanisms:

- No authentication
- Plain text authentication

VRRP rejects packets in any of the following cases:

- The authentication schemes differ on the router and in the incoming packet.
- Text authentication strings differ on the router and in the incoming packet.

VRRPv3

VRRP version 3 (VRRPv3) enables a group of switches to form a single virtual switch in order to provide redundancy and reduce the possibility of a single point of failure in a network. The LAN clients can then be configured with the virtual switch as their default gateway. The virtual switch, representing a group of switches, is also known as a VRRPv3 group.

Virtualization Support

VRRP supports virtual routing and forwarding (VRF) instances. By default, Cisco NX-OS places you in the default VRF unless you specifically configure another VRF.

If you change the VRF membership of an interface, Cisco NX-OS removes all Layer 3 configuration, including VRRP.

For more information, see [Chapter 14, “Configuring Layer 3 Virtualization.”](#)

Licensing Requirements for VRRP

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	VRRP requires 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. For a complete explanation of the Cisco NX-OS licensing scheme, see the <i>Cisco NX-OS Licensing Guide</i> .
	Note Make sure the LAN Base Services license is installed on the switch to enable Layer 3 interfaces.

Guidelines and Limitations

VRRP has the following configuration guidelines and limitations:

- You cannot configure VRRP on the management interface.
- When VRRP is enabled, you should replicate the VRRP configuration across switches in your network.
- We recommend that you do not configure more than one first-hop redundancy protocol on the same interface.
- You must configure an IP address for the interface that you configure VRRP on and enable that interface before VRRP becomes active.
- Cisco NX-OS removes all Layer 3 configurations on an interface when you change the interface VRF membership, port channel membership, or when you change the port mode to Layer 2.
- When you configure VRRP to track a Layer 2 interface, you must shut down the Layer 2 interface and reenabling the interface to update the VRRP priority to reflect the state of the Layer 2 interface.
- VRRPv3 has the following configuration guidelines and limitations:
 - VRRPv3 is not intended as a replacement for existing dynamic protocols. VRRPv3 is designed for use over multi-access, multicast, or broadcast-capable Ethernet LANs.
 - VRRPv3 is supported only on Ethernet and Fast Ethernet interfaces, bridge group virtual interfaces (BVs), and Gigabit Ethernet interfaces as well as on Multiprotocol Label Switching (MPLS) virtual private networks (VPNs), VRF-aware MPLS VPNs, and VLANs.
 - When VRRPv3 is in use, VRRPv2 is unavailable. To configure VRRPv3, you must disable any VRRPv2 configuration.
 - Use VRRPv3 millisecond timers only where absolutely necessary and with careful consideration and testing. Millisecond values work only under favorable circumstances. The millisecond timer values are compatible with third-party vendors, as long as they also support VRRPv3.

Default Settings

Table 21-1 lists the default settings for VRRP parameters.

Table 21-1 *Default VRRP Parameters*

Parameters	Default
advertisement interval	1 second
authentication	No authentication
preemption	Enabled
priority	100
VRRP feature	Disabled
VRRPv3	Disabled
VRRPv3 secondary address matching	Enables
VRRPv3 timers advertise	1000 milliseconds

Configuring VRRP

This section includes the following topics:

- [Enabling the VRRP Feature, page 21-7](#)
- [Configuring VRRP Groups, page 21-7](#)
- [Configuring VRRP Priority, page 21-9](#)
- [Configuring VRRP Authentication, page 21-10](#)
- [Configuring Time Intervals for Advertisement Packets, page 21-12](#)
- [Disabling Preemption, page 21-13](#)
- [Configuring VRRP Interface State Tracking, page 21-15](#)

**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 the VRRP Feature

You must globally enable the VRRP feature before you can configure and enable any VRRP groups.

To enable the VRRP feature, use the following command in global configuration mode:

Command	Purpose
feature vrrp	Enables VRRP.
Example: switch(config)# feature vrrp	

To disable the VRRP feature and remove all associated configuration, use the following command in global configuration mode:

Command	Purpose
no feature vrrp	Disables the VRRP feature.
Example: switch(config)# no feature vrrp	

Configuring VRRP Groups

You can create a VRRP group, assign the virtual IP address, and enable the group.

You can configure one virtual IPv4 address for a VRRP group. By default, the master VRRP router drops the packets addressed directly to the virtual IP address because the VRRP master is only intended as a next-hop router to forward packets. Some applications require that Cisco NX-OS accept packets addressed to the virtual router IP. Use the secondary option to the virtual IP address to accept these packets when the local router is the VRRP master.

Once you have configured the VRRP group, you must explicitly enable the group before it becomes active.

BEFORE YOU BEGIN

Ensure that you configure an IP address on the interface (see the [“Configuring IPv4 Addressing”](#) section on page 2-8).

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **no switchport**
4. **vrrp** *number*
5. **address** *ip-address* [secondary]
6. **no shutdown**
7. (Optional) **show vrrp**
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# switch(config-if)# interface ethernet 1/2	Enters interface configuration mode.
Step 3	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	vrrp <i>number</i> Example: switch(config-if)# vrrp 250 switch(config-if-vrrp)#	Creates a virtual router group. The range is from 1 to 255.

	Command	Purpose
Step 5	address <i>ip-address</i> [secondary] Example: switch(config-if-vrrp)# address 192.0.2.8	Configures the virtual IPv4 address for the specified VRRP group. This address should be in the same subnet as the IPv4 address of the interface. Use the secondary option only if applications require that VRRP routers accept the packets sent to the virtual router's IP address and deliver to applications.
Step 6	no shutdown Example: switch(config-if-vrrp)# no shutdown switch(config-if-vrrp)#	Enables the VRRP group. Disabled by default.
Step 7	show vrrp Example: switch(config-if-vrrp)# show vrrp	(Optional) Displays VRRP information.
Step 8	copy running-config startup-config Example: switch(config-if-vrrp)# copy running-config startup-config	(Optional) Saves this configuration change.

Configuring VRRP Priority

The valid priority range for a virtual router is from 1 to 254 (1 is the lowest priority and 254 is the highest). The default priority value for backups is 100. For switches whose interface IP address is the same as the primary virtual IP address (the master), the default value is 255.

BEFORE YOU BEGIN

Ensure that you have enabled the VRRP feature (see the [“Configuring VRRP”](#) section on page 21-7).

Ensure that you have configured an IP address on the interface (see the [“Configuring IPv4 Addressing”](#) section on page 2-8).

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **no switchport**
4. **vrrp** *number*
5. **shutdown**
6. **priority** *level* [**forwarding-threshold** *lower lower-value upper upper-value*]
7. **no shutdown**
8. (Optional) **show vrrp**

9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface interface-type slot/port Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	vrrp number Example: switch(config-if)# vrrp 250 switch(config-if-vrrp)#	Creates a virtual router group.
Step 5	shutdown Example: switch(config-if-vrrp)# shutdown switch(config-if-vrrp)#	Disables the VRRP group. Disabled by default.
Step 6	priority level [forwarding-threshold lower lower-value upper upper-value] Example: switch(config-if-vrrp)# priority 60 forwarding-threshold lower 40 upper 50	Sets the priority level used to select the active router in an VRRP group. The <i>level</i> range is from 1 to 254. The default is 100 for backups and 255 for a master that has an interface IP address equal to the virtual IP address.
Step 7	no shutdown Example: switch(config-if-vrrp)# no shutdown switch(config-if-vrrp)#	Enables the VRRP group. Disabled by default.
Step 8	show vrrp Example: switch(config-if-vrrp)# show vrrp	(Optional) Displays a summary of VRRP information.
Step 9	copy running-config startup-config Example: switch(config-if-vrrp)# copy running-config startup-config	(Optional) Saves this configuration change.

Configuring VRRP Authentication

You can configure simple text authentication for a VRRP group.

BEFORE YOU BEGIN

Ensure that the authentication configuration is identical for all VRRP switches in the network.

Ensure that you have enabled the VRRP feature (see the “Configuring VRRP” section on page 21-7).

Ensure that you have configured an IP address on the interface (see the “Configuring IPv4 Addressing” section on page 2-8).

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **no switchport**
4. **vrrp** *number*
5. **shutdown**
6. **authentication text** *password*
7. **no shutdown**
8. (Optional) **show vrrp**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	vrrp <i>number</i> Example: switch(config-if)# vrrp 250 switch(config-if-vrrp)#	Creates a virtual router group.
Step 5	shutdown Example: switch(config-if-vrrp)# shutdown switch(config-if-vrrp)#	Disables the VRRP group. Disabled by default.

	Command	Purpose
Step 6	authentication text <i>password</i> Example: switch(config-if-vrrp)# authentication md5 prd555o1n47espn0 spi 0x0	Assigns the simple text authentication option and specifies the keyname password. The keyname range is from 1 to 255 characters. We recommend that you use at least 16 characters. The text password is up to eight alphanumeric characters.
Step 7	no shutdown Example: switch(config-if-vrrp)# no shutdown switch(config-if-vrrp)#	Enables the VRRP group. Disabled by default.
Step 8	show vrrp Example: switch(config-if-vrrp)# show vrrp	(Optional) Displays a summary of VRRP information.
Step 9	copy running-config startup-config Example: switch(config-if-vrrp)# copy running-config startup-config	(Optional) Saves this configuration change.

Configuring Time Intervals for Advertisement Packets

You can configure the time intervals for advertisement packets.

BEFORE YOU BEGIN

Ensure that you have enabled the VRRP feature (see the [“Configuring VRRP”](#) section on page 21-7).

Ensure that you have configured an IP address on the interface (see the [“Configuring IPv4 Addressing”](#) section on page 2-8).

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **no switchport**
4. **vrrp** *number*
5. **shutdown**
6. **advertisement-interval** *seconds*
7. **no shutdown**
8. (Optional) **show vrrp**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	vrrp number Example: switch(config-if)# vrrp 250 switch(config-if-vrrp)#	Creates a virtual router group.
Step 5	shutdown Example: switch(config-if-vrrp)# shutdown switch(config-if-vrrp)#	Disables the VRRP group. Disabled by default.
Step 6	advertisement-interval <i>seconds</i> Example: switch(config-if-vrrp)# advertisement-interval 15	Sets the interval time in seconds between sending advertisement frames. The range is from 1 to 254. The default is 1 second.
Step 7	no shutdown Example: switch(config-if-vrrp)# no shutdown switch(config-if-vrrp)#	Enables the VRRP group. Disabled by default.
Step 8	show vrrp Example: switch(config-if-vrrp)# show vrrp	(Optional) Displays a summary of VRRP information.
Step 9	copy running-config startup-config Example: switch(config-if-vrrp)# copy running-config startup-config	(Optional) Saves this configuration change.

Disabling Preemption

You can disable preemption for a VRRP group member. If you disable preemption, a higher-priority backup router will not take over for a lower-priority master router. Preemption is enabled by default.

BEFORE YOU BEGIN

Ensure that you have enabled the VRRP feature (see the [“Configuring VRRP” section on page 21-7](#)).

Ensure that you have configured an IP address on the interface (see the [“Configuring IPv4 Addressing” section on page 2-8](#)).

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **no switchport**
4. **vrrp** *number*
5. **shutdown**
6. **no preempt**
7. **no shutdown**
8. (Optional) **show vrrp**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	vrrp <i>number</i> Example: switch(config-if)# vrrp 250 switch(config-if-vrrp)#	Creates a virtual router group.
Step 5	no shutdown Example: switch(config-if-vrrp)# no shutdown	Enables the VRRP group. Disabled by default.
Step 6	no preempt Example: switch(config-if-vrrp)# no preempt	Disables the preempt option and allows the master to remain when a higher-priority backup appears.

	Command	Purpose
Step 7	no shutdown Example: switch(config-if-vrrp)# no shutdown	Enables the VRRP group. Disabled by default.
Step 8	show vrrp Example: switch(config-if-vrrp)# show vrrp	(Optional) Displays a summary of VRRP information.
Step 9	copy running-config startup-config Example: switch(config-if-vrrp)# copy running-config startup-config	(Optional) Saves this configuration change.

Configuring VRRP Interface State Tracking

Interface state tracking changes the priority of the virtual router based on the state of another interface in the switch. When the tracked interface goes down or the IP address is removed, Cisco NX-OS assigns the tracking priority value to the virtual router. When the tracked interface comes up and an IP address is configured on this interface, Cisco NX-OS restores the configured priority to the virtual router (see the [“Configuring VRRP Priority” section on page 21-9](#)).



Note

For interface state tracking to function, you must enable preemption on the interface.



Note

VRRP does not support Layer 2 interface tracking.

BEFORE YOU BEGIN

Ensure that you have enabled the VRRP feature (see the [“Configuring VRRP” section on page 21-7](#)).

Ensure that you have configured an IP address on the interface (see the [“Configuring IPv4 Addressing” section on page 2-8](#)).

Ensure that you have enabled the virtual router (see the [“Configuring VRRP Groups” section on page 21-7](#)).

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **no switchport**
4. **vrrp** *number*
5. **shutdown**
6. **track interface** *type number priority value*
7. **no shutdown**
8. (Optional) **show vrrp**

9. (Optional) copy running-config startup-config

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface interface-type slot/port Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	no switchport Example: switch(config-if)# no switchport	Configures the interface as a Layer 3 routed interface.
Step 4	vrrp number Example: switch(config-if)# vrrp 250 switch(config-if-vrrp)#	Creates a virtual router group.
Step 5	shutdown Example: switch(config-if-vrrp)# shutdown switch(config-if-vrrp)#	Disables the VRRP group. Disabled by default.
Step 6	track interface type number priority value Example: switch(config-if-vrrp)# track interface ethernet 2/10 priority 254	Enables interface priority tracking for a VRRP group. The priority range is from 1 to 254.
Step 7	no shutdown Example: switch(config-if-vrrp)# no shutdown switch(config-if-vrrp)#	Enables the VRRP group. Disabled by default.
Step 8	show vrrp Example: switch(config-if-vrrp)# show vrrp	(Optional) Displays a summary of VRRP information.
Step 9	copy running-config startup-config Example: switch(config-if-vrrp)# copy running-config startup-config	(Optional) Saves this configuration change.

Configuring VRRPv3

Enabling VRRPv3

You must globally enable the VRRPv3 feature before you can configure and enable any VRRPv3 groups. To enable the VRRPv3 feature, use the following command in global configuration mode:

Command	Purpose
feature vrrpv3	Enables VRRP version 3. The no form of this command disables VRRPv3 in a VDC.
Example: switch(config)# feature vrrpv3	If VRRPv2 is currently configured, use the no feature vrrp command in global configuration mode to remove the VRRPv2 configuration and then use the feature vrrpv3 command to enable VRRPv3.

Configuring VRRPv3 Groups

You can create a VRRPv3 group, assign the virtual IP address, and enable the group.

You can configure one virtual IPv4 address for a VRRPv3 group. By default, the master VRRPv3 router drops the packets addressed directly to the virtual IP address because the VRRPv3 master is only intended as a next-hop router to forward packets. Some applications require that Cisco NX-OS accept packets addressed to the virtual router IP. Use the secondary option to the virtual IP address to accept these packets when the local router is the VRRPv3 master.



Note

After you have configured the VRRPv3 group, you must explicitly enable the group before it becomes active.

BEFORE YOU BEGIN

- Ensure that the VRRPv3 feature is enabled.
- Ensure that you configure an IP address on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrpv3** *number* **address-family** {**ipv4** | **ipv6**}
4. (Optional) **address** *ip-address* [**primary** | **secondary**]
5. (Optional) **description** *description*
6. (Optional) **match-address**
7. (Optional) **preempt** [**delay minimum** *seconds*]
8. (Optional) **priority** *level*

9. (Optional) **timers advertise interval**
10. (Optional) **vrrpv2**
11. (Optional) **shutdown**
12. (Optional) **show fhrp** [*interface-type interface-number*] [**verbose**]
13. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 2/1 switch(config-if)#	Enters interface configuration mode.
Step 3	[no] vrrpv3 <i>number address-family {ipv4 ipv6}</i> Example: switch(config-if)# vrrpv3 5 address-family ipv4 switch(config-if-vrrpv3-group)#	Creates a VRRPv3 group and enters VRRPv3 group configuration mode. The range is from 1 to 255. The no form of this command removes all configuration defined within the submode.
Step 4	[no] address <i>ip-address [primary secondary]</i> Example: switch(config-if-vrrpv3-group)# address 100.0.1.10 primary	(Optional) Specifies a primary or secondary IPv4 or IPv6 address for the VRRPv3 group. Note To utilize secondary IP addresses in a VRRPv3 group, you must first configure a primary IP address on the same group.
Step 5	[no] description <i>description</i> Example: switch(config-if-vrrpv3-group)# description group3	(Optional) Specifies a description for the VRRPv3 group. You can enter up to 80 alphanumeric characters.
Step 6	[no] match-address Example: switch(config-if-vrrpv3-group)# match-address	(Optional) Matches the secondary address in the advertisement packet against the configured address.
Step 7	[no] preempt [delay minimum seconds] Example: switch(config-if-vrrpv3-group)# preempt delay minimum 30	(Optional) Enables preemption of a lower priority master switch with an optional delay. The range is from 0 to 3600.
Step 8	[no] priority <i>level</i> Example: switch(config-if-vrrpv3-group)# priority 3	(Optional) Specifies the priority of the VRRPv3 group. The range is from 1 to 254.

	Command	Purpose
Step 9	<pre>[no] timers advertise interval</pre> <p>Example:</p> <pre>switch(config-if-vrrpv3-group)# timers advertise 1000</pre>	<p>(Optional) Sets the advertisement timer in milliseconds. The range is from 100 to 40950.</p> <p>Note Cisco recommends that you set this timer to a value greater than or equal to 1 second.</p>
Step 10	<pre>[no] vrrpv2</pre> <p>Example:</p> <pre>switch(config-if-vrrpv3-group)# vrrpv2</pre>	<p>(Optional) Enables support for VRRPv2 simultaneously, to ensure interoperability with devices that support only VRRPv2.</p> <p>Note VRRPv2 compatibility mode is provided to allow an upgrade from VRRPv2 to VRRPv3. This is not a full VRRPv2 implementation and should be used only to perform an upgrade.</p>
Step 11	<pre>[no] shutdown</pre> <p>Example:</p> <pre>switch(config-if-vrrpv3-group)# shutdown</pre>	<p>(Optional) Disables VRRP configuration for the VRRPv3 group.</p>
Step 12	<pre>show fhrp [interface-type interface-number] [verbose]</pre> <p>Example:</p> <pre>switch(config-if-vrrpv3-group)# show fhrp port-channel 101 verbose</pre>	<p>(Optional) Displays First Hop Redundancy Protocol (FHRP) information.</p> <p>Use the verbose keyword to view detailed information.</p>
Step 13	<pre>copy running-config startup-config</pre> <p>Example:</p> <pre>switch(config-if-vrrpv3-group)# copy running-config startup-config</pre>	<p>(Optional) Saves this configuration change.</p>

Configuring the Delay Period for FHRP Client Initialization

You can configure the delay period for the initialization of FHRP clients.



Note

In all FHRP protocols, we do not recommend to use aggressive timers as they cause CPU spikes and they result in increased control packet flow. In case of VRRPv3, you should configure sufficient interface delay/reload delay for proper failover of the VRRP nodes.

To configure this feature, use the following command in interface configuration mode:

Command	Purpose
fhrp delay {[minimum] [reload] <i>seconds</i> } Example: switch(config)# fhrp delay minimum 34	Specifies the delay period for the initialization of FHRP clients. The range is from 0 to 3600 seconds. The minimum keyword configures the delay period after an interface becomes available. The reload command configures the delay period after the device reloads.

Configuring VRRPv3 Control Groups

You can configure a VRRPv3 control group.

BEFORE YOU BEGIN

- Ensure that the VRRPv3 feature is enabled.
- Ensure that you configure an IP address on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ip address** *ip-address mask* [**secondary**]
4. **vrrpv3 number address-family** {**ipv4** | **ipv6**}
5. (Optional) **address** *ip-address* [**primary** | **secondary**]
6. (Optional) **shutdown**
7. (Optional) **show fhrp** [*interface-type interface-number*] [**verbose**]
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 2/1 switch(config-if)#	Enters interface configuration mode.

	Command	Purpose
Step 3	<pre>[no] ip address ip-address mask [secondary]</pre> <p>Example: switch(config-if)# ip address 209.165.200.230 255.255.255.224</p>	<p>Configures the IP address on the interface.</p> <p>Note You can use the secondary keyword to configure additional IP addresses on the interface.</p>
Step 4	<pre>[no] vrrpv3 number address-family {ipv4 ipv6}</pre> <p>Example: switch(config-if)# vrrpv3 5 address-family ipv4 switch(config-if-vrrpv3-group)#</p>	Creates a VRRPv3 group and enters VRRPv3 group configuration mode. The range is from 1 to 255.
Step 5	<pre>[no] address ip-address [primary secondary]</pre> <p>Example: switch(config-if-vrrpv3-group)# address 209.165.200.227 primary</p>	(Optional) Specifies a primary or secondary IPv4 or IPv6 address for the VRRPv3 group.
Step 6	<pre>[no] shutdown</pre> <p>Example: switch(config-if-vrrpv3-group)# shutdown</p>	(Optional) Disables VRRP configuration for the VRRPv3 group.
Step 7	<pre>show fhrp [interface-type interface-number] [verbose]</pre> <p>Example: switch(config-if-vrrpv3-group)# show fhrp port-channel 101 verbose</p>	<p>(Optional) Displays First Hop Redundancy Protocol (FHRP) information.</p> <p>Use the verbose keyword to view detailed information.</p>
Step 8	<pre>copy running-config startup-config</pre> <p>Example: switch(config-if-vrrpv3-group)# copy running-config startup-config</p>	(Optional) Saves this configuration change.

Verifying the VRRPv2 Configuration

To display the VRRPv2 configuration information, perform one of the following tasks:

Command	Purpose
show vrrpv2	Displays the VRRP status for all groups.
show vrrpv2 vr group-number	Displays the VRRP status for a VRRP group.
show vrrpv2 vr number interface interface-type port configuration	Displays the virtual router configuration for an interface.
show vrrpv2 vr number interface interface-type port status	Displays the virtual router status for an interface.
show fhrp [interface-type interface-number] [verbose]	Displays First Hop Redundancy Protocol (FHRP) information.
show interface interface-type	Displays the virtual router configuration for an interface.

Verifying the VRRPv3 Configuration

See the following table for information on the fields in the **show vrrpv3** command output:

Command	Purpose
>	Redirect it to a file
>>	Redirect it to a file in append mode
all	All VRRPV3 information
brief	Brief output
detail	Detail output
ethernet	Ethernet IEEE 802.3z
internal	Internal command
ipv4	IPv4
ipv6	IPv6
loopback	Loopback interface
port-channel	Port Channel interface
statistics	Statistics output
Vlan	VLAN interface
	Pipe command output to filter

For example, use the **show vrrpv3 statistics** command to display the VRRPv3 statistics:

```
# show vrrpv3 statistics
```

Displaying VRRP Statistics

To display VRRP statistics, use the following commands:

Command	Purpose
show vrrp vr <i>number</i> interface <i>interface-type</i> <i>port</i> statistics	Displays the virtual router information.
show vrrp statistics	Displays the VRRP statistics.

Use the **clear vrrp vr** command to clear the IPv4 VRRP statistics for a specified interface.

Use the **clear vrrp ipv4** command to clear all the statistics for the specified IPv4 virtual router.

Configuration Examples for VRRPv2

In this example, Router A and Router B each belong to three VRRPv2 groups. In the configuration, each group has the following properties:

- Group 1:
 - Virtual IP address is 10.1.0.10.
 - Router A will become the master for this group with priority 120.
 - Advertising interval is 3 seconds.
 - Preemption is enabled.
- Group 5:
 - Router B will become the master for this group with priority 200.
 - Advertising interval is 30 seconds.
 - Preemption is enabled.
- Group 100:
 - Router A will become the master for this group first because it has a higher IP address (10.1.0.2).
 - Advertising interval is the default 1 second.
 - Preemption is disabled.

Router A

```
interface ethernet 1/0
  no switchport
  ip address 10.1.0.2/16
  no shutdown
  vrrpv2 1
    priority 120
    authentication text cisco
    advertisement-interval 3
    address 10.1.0.10
    no shutdown
  vrrpv2 5
    priority 100
    advertisement-interval 30
    address 10.1.0.50
    no shutdown
  vrrpv2 100
    no preempt
    address 10.1.0.100
    no shutdown
```

Router B

```
interface ethernet 1/0
  no switchport
  ip address 10.2.0.1/2
  no shutdown
  vrrpv2 1
    priority 100
    authentication text cisco
    advertisement-interval 3
    address 10.2.0.10
    no shutdown

  vrrpv2 5
    priority 200
    advertisement-interval 30
    address 10.2.0.50
    no shutdown
  vrrpv2 100
    no preempt
```

```
address 10.2.0.100
no shutdown
```

Configuration Example for VRRPv3

See the following configuration example for VRRPv3.

```
interface Vlan20
  vrrpv3 10 address-family ipv4
    timers advertise 1000
    priority 100
    preempt
    match-address
    no vrrpv2
    address 20.1.1.1 primary
    address 20.1.1.5 secondary
  vrrpv3 10 address-family ipv6
    timers advertise 1000
    priority 100
    preempt
    match-address
    no vrrpv2
    address fe80::1 primary
    address 2011::5
```

Additional References

For additional information related to implementing VRRP, see the following sections:

- [Related Documents, page 21-24](#)

Related Documents

Related Topic	Document Title
Configuring the Hot Standby Routing Protocol	Chapter 20, “Configuring HSRP”
VRRP CLI commands	<i>Cisco Nexus 3000 Series Command Reference</i>



Configuring Object Tracking

This chapter describes how to configure object tracking on Cisco NX-OS switches.

This chapter includes the following sections:

- [Information About Object Tracking, page 22-1](#)
- [Licensing Requirements for Object Tracking, page 22-3](#)
- [Guidelines and Limitations, page 22-3](#)
- [Default Settings, page 22-3](#)
- [Configuring Object Tracking, page 22-3](#)
- [Verifying the Object Tracking Configuration, page 22-13](#)
- [Configuration Examples for Object Tracking, page 22-13](#)
- [Related Topics, page 22-13](#)
- [Additional References, page 22-13](#)

Information About Object Tracking

Object tracking allows you to track specific objects on the switch, such as the interface line protocol state, IP routing, and route reachability, and to take action when the tracked object's state changes. This feature allows you to increase the availability of the network and shorten recovery time if an object state goes down.

This section includes the following topics:

- [Object Tracking Overview, page 22-1](#)
- [Object Track List, page 22-2](#)
- [Virtualization Support, page 22-2](#)

Object Tracking Overview

The object tracking feature allows you to create a tracked object that multiple clients can use to modify the client behavior when a tracked object changes. Several clients register their interest with the tracking process, track the same object, and take different actions when the object state changes.

Clients include the following features:

- Hot Standby Redundancy Protocol (HSRP)

- Virtual Router Redundancy Protocol (VRRP)
- Embedded Event Manager (EEM)

The object tracking monitors the status of the tracked objects and communicates any changes made to interested clients. Each tracked object is identified by a unique number that clients can use to configure the action to take when a tracked object changes state.

Cisco NX-OS tracks the following object types:

- Interface line protocol state—Tracks whether the line protocol state is up or down.
- Interface IP routing state—Tracks whether the interface has an IPv4 address and if IPv4 routing is enabled and active.
- IP route reachability—Tracks whether an IPv4 route exists and is reachable from the local switch.

For example, you can configure HSRP to track the line protocol of the interface that connects one of the redundant routers to the rest of the network. If that link protocol goes down, you can modify the priority of the affected HSRP router.

Object Track List

An object track list allows you to track the combined states of multiple objects. Object track lists support the following capabilities:

- Boolean "and" function—Each object defined within the track list must be in an up state so that the track list object can become up.
- Boolean "or" function—At least one object defined within the track list must be in an up state so that the tracked object can become up.
- Threshold percentage—The percentage of up objects in the tracked list must be greater than the configured up threshold for the tracked list to be in the up state. If the percentage of down objects in the tracked list is above the configured track list down threshold, the tracked list is marked as down.
- Threshold weight—Assign a weight value to each object in the tracked list, and a weight threshold for the track list. If the combined weights of all up objects exceeds the track list weight up threshold, the track list is in an up state. If the combined weights of all the down objects exceeds the track list weight down threshold, the track list is in the down state.

See the [“Configuring an Object Track List with a Boolean Expression”](#) section on page 22-6 for more information on track lists.

Virtualization Support

Object tracking supports virtual routing and forwarding (VRF) instances. By default, Cisco NX-OS places you in the default VRF unless you specifically configure another VRF. By default, Cisco NX-OS tracks the route reachability state of objects in the default VRF. If you want to track objects in another VRF, you must configure the object to be a member of that VRF (see the [“Configuring Object Tracking for a Nondefault VRF”](#) section on page 22-12).

For more information, see [Chapter 14, “Configuring Layer 3 Virtualization.”](#)

Licensing Requirements for Object Tracking

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	Object tracking requires 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. For a complete explanation of the Cisco NX-OS licensing scheme, see the <i>Cisco NX-OS Licensing Guide</i> .

Guidelines and Limitations

Object tracking has the following configuration guidelines and limitations:

- Supports up to 500 tracked objects.
- Supports Ethernet, subinterfaces, tunnels, port channels, loopback interfaces, and VLAN interfaces.
- Supports one tracked object per HSRP group.

Default Settings

Table 22-1 lists the default settings for object tracking parameters.

Table 22-1 Default Object Tracking Parameters

Parameters	Default
Tracked Object VRF	Member of default VRF

Configuring Object Tracking

This section includes the following topics:

- [Configuring Object Tracking for an Interface, page 22-4](#)
- [Configuring Object Tracking for Route Reachability, page 22-5](#)
- [Configuring an Object Track List with a Boolean Expression, page 22-6](#)
- [Configuring an Object Track List with a Percentage Threshold, page 22-7](#)
- [Configuring an Object Track List with a Weight Threshold, page 22-8](#)
- [Configuring an Object Tracking Delay, page 22-10](#)
- [Configuring Object Tracking for a Nondefault VRF, page 22-12](#)



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.

Configuring Object Tracking for an Interface

You can configure Cisco NX-OS to track the line protocol or IPv4 routing state of an interface.

SUMMARY STEPS

1. **configure terminal**
2. **track object-id interface interface-type number {ip routing | line-protocol}**
3. (Optional) **show track [object-id]**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	track object-id interface interface-type number {ip routing line-protocol} Example: switch(config)# track 1 interface ethernet 1/2 line-protocol switch(config-track)#	Creates a tracked object for an interface and enters tracking configuration mode. The <i>object-id</i> range is from 1 to 500.
Step 3	show track [object-id] Example: switch(config-track)# show track 1	(Optional) Displays object tracking information.
Step 4	copy running-config startup-config Example: switch(config-track)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure object tracking for the line protocol state on Ethernet 1/2:

```
switch# configure terminal
switch(config)# track 1 interface ethernet 1/2 line-protocol
switch(config-track)# copy running-config startup-config
```

This example shows how to configure object tracking for the IPv4 routing state on Ethernet 1/2:

```
switch# configure terminal
switch(config)# track 2 interface ethernet 1/2 ip routing
switch(config-track)# copy running-config startup-config
```

Configuring Object Tracking for Route Reachability

You can configure Cisco NX-OS to track the existence and reachability of an IP route.

SUMMARY STEPS

1. **configure terminal**
2. **track *object-id* ip route *prefix/length* reachability**
3. (Optional) **show track [*object-id*]**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	track <i>object-id</i> ip route <i>prefix/length</i> reachability Example: switch(config)# track 2 ip route 192.0.2.0/8 reachability switch(config-track)#	Creates a tracked object for a route and enters tracking configuration mode. The <i>object-id</i> range is from 1 to 500. The prefix format for IP is A.B.C.D/length, where the length range is from 1 to 32.
Step 3	show track [<i>object-id</i>] Example: switch(config-track)# show track 1	(Optional) Displays object tracking information.
Step 4	copy running-config startup-config Example: switch(config-track)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure object tracking for an IPv4 route in the default VRF:

```
switch# configure terminal
switch(config)# track 4 ip route 192.0.2.0/8 reachability
switch(config-track)# copy running-config startup-config
```

Configuring an Object Track List with a Boolean Expression

You can configure an object track list that contains multiple tracked objects. A tracked list contains one or more objects. The Boolean expression enables two types of calculation by using either "and" or "or" operators. For example, when tracking two interfaces using the "and" operator, up means that both interfaces are up, and down means that either interface is down.

SUMMARY STEPS

1. **configure terminal**
2. **track** *track-number* **list boolean** {**and** | **or**}
3. **object** *object-number* [**not**]
4. (Optional) **show track**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	track <i>track-number</i> list boolean { and or } Example: switch(config)# track 1 list boolean and switch(config-track)#	Configures a tracked list object and enters tracking configuration mode. Specifies that the state of the tracked list is based on a Boolean calculation. The keywords are as follows: <ul style="list-style-type: none"> • and—Specifies that the list is up if all objects are up, or down if one or more objects are down. For example, when tracking two interfaces, up means that both interfaces are up, and down means that either interface is down. • or—Specifies that the list is up if at least one object is up. For example, when tracking two interfaces, up means that either interface is up, and down means that both interfaces are down. The <i>track-number</i> range is from 1 to 500.
Step 3	object <i>object-id</i> [not] Example: switch(config-track)# object 10	Adds a tracked object to the track list. The <i>object-id</i> range is from 1 to 500. The not keyword optionally negates the tracked object state. Note The example means that when object 10 is up, the tracked list detects object 10 as down.

	Command	Purpose
Step 4	show track Example: switch(config-track)# show track	(Optional) Displays object tracking information.
Step 5	copy running-config startup-config Example: switch(config-track)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure a track list with multiple objects as a Boolean “and”:

```
switch# configure terminal
switch(config)# track 1 list boolean and
switch(config-track)# object 10
switch(config-track)# object 20 not
```

Configuring an Object Track List with a Percentage Threshold

You can configure an object track list that contains a percentage threshold. A tracked list contains one or more objects. The percentage of up objects must exceed the configured track list up percent threshold before the track list is in an up state. For example, if the tracked list has three objects, and you configure an up threshold of 60 percent, two of the objects must be in the up state (66 percent of all objects) for the track list to be in the up state.

SUMMARY STEPS

1. **configure terminal**
2. **track** *track-number* **list threshold percentage**
3. **threshold percentage up** *up-value* **down** *down-value*
4. **object** *object-number*
5. (Optional) **show track**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	track <i>track-number</i> list threshold percentage Example: switch(config)# track 1 list threshold percentage switch(config-track)#	Configures a tracked list object and enters tracking configuration mode. Specifies that the state of the tracked list is based on a configured threshold percent. The <i>track-number</i> range is from 1 to 500.

	Command	Purpose
Step 3	threshold percentage up <i>up-value</i> down <i>down-value</i> Example: switch(config-track)# threshold percentage up 70 down 30	Configures the threshold percent for the tracked list. The range from 0 to 100 percent.
Step 4	object <i>object-id</i> Example: switch(config-track)# object 10	Adds a tracked object to the track list. The <i>object-id</i> range is from 1 to 500.
Step 5	show track Example: switch(config-track)# show track	(Optional) Displays object tracking information.
Step 6	copy running-config startup-config Example: switch(config-track)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure a track list with an up threshold of 70 percent and a down threshold of 30 percent:

```
switch# configure terminal
switch(config)# track 1 list threshold percentage
switch(config-track)# threshold percentage up 70 down 30
switch(config-track)# object 10
switch(config-track)# object 20
switch(config-track)# object 30
```

Configuring an Object Track List with a Weight Threshold

You can configure an object track list that contains a weight threshold. A tracked list contains one or more objects. The combined weight of up objects must exceed the configured track list up weight threshold before the track list is in an up state. For example, if the tracked list has three objects with the default weight of 10 each, and you configure an up threshold of 15, two of the objects must be in the up state (combined weight of 20) for the track list to be in the up state.

SUMMARY STEPS

1. **configure terminal**
2. **track** *track-number* **list threshold weight**
3. **threshold weight up** *up-value* **down** *down-value*
4. **object** *object-number* **weight** *value*
5. (Optional) **show track**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	track track-number list threshold weight Example: switch(config)# track 1 list threshold weight switch(config-track)#	Configures a tracked list object and enters tracking configuration mode. Specifies that the state of the tracked list is based on a configured threshold weight. The <i>track-number</i> range is from 1 to 500.
Step 3	threshold weight up up-value down down-value Example: switch(config-track)# threshold weight up 30 down 10	Configures the threshold weight for the tracked list. The range from 1 to 255.
Step 4	object object-id weight value Example: switch(config-track)# object 10 weight 15	Adds a tracked object to the track list. The <i>object-id</i> range is from 1 to 500. The <i>value</i> range is from 1 to 255. The default weight value is 10.
Step 5	show track Example: switch(config-track)# show track	(Optional) Displays object tracking information.
Step 6	copy running-config startup-config Example: switch(config-track)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure a track list with an up weight threshold of 30 and a down threshold of 10:

```
switch# configure terminal
switch(config)# track 1 list threshold weight
switch(config-track)# threshold weight up 30 down 10
switch(config-track)# object 10 weight 15
switch(config-track)# object 20 weight 15
switch(config-track)# object 30
```

In this example, the track list is up if object 10 and object 20 are up, and the track list goes to the down state if all three objects are down.

Configuring an Object Tracking Delay

You can configure a delay for a tracked object or an object track list that delays when the object or list triggers a stage change. The tracked object or track list starts the delay timer when a state change occurs but does not recognize a state change until the delay timer expires. At that point, Cisco NX-OS checks the object state again and records a state change only if the object or list currently has a changed state. Object tracking ignores any intermediate state changes before the delay timer expires.

For example, for an interface line-protocol tracked object that is in the up state with a 20-second down delay, the delay timer starts when the line protocol goes down. The object is not in the down state unless the line protocol is down 20 seconds later.

You can configure independent up delay and down delay for a tracked object or track list. When you delete the delay, object tracking deletes both the up and down delay.

You can change the delay at any point. If the object or list is already counting down the delay timer from a triggered event, the new delay is computed as the following:

- If the new configuration value is less than the old configuration value, the timer starts with the new value.
- If the new configuration value is more than the old configuration value, the timer is calculated as the new configuration value minus the current timer countdown minus the old configuration value.

SUMMARY STEPS

1. **configure terminal**
2. **track** *object-id* {*parameters*}
3. **track** *track-number list* {*parameters*}
4. **delay** {**up** *up-time* [**down** *down-time*] | **down** *down-time* [**up** *up-time*]}
5. (Optional) **show track**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	track <i>object-id</i> { <i>parameters</i> }	Creates a tracked object for a route and enters tracking configuration mode. The <i>object-id</i> range is from 1 to 500. The prefix format for IP is A.B.C.D/length, where the length range is from 1 to 32.
Step 3	track <i>track-number list</i> { <i>parameters</i> }	Configures a tracked list object and enters tracking configuration mode. Specifies that the state of the tracked list is based on a configured threshold weight. The <i>track-number</i> range is from 1 to 500.
	Example: switch(config)# track 1 list threshold weight switch(config-track)#	

	Command	Purpose
Step 4	delay { up <i>up-time</i> [down <i>down-time</i>] down <i>down-time</i> [up <i>up-time</i>] } Example: switch(config-track)# delay up 20 down 30	Configures the object delay timers. The range is from 0 to 180 seconds.
Step 5	show track Example: switch(config-track)# show track 3	(Optional) Displays object tracking information.
Step 6	copy running-config startup-config Example: switch(config-track)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure object tracking for a route and use delay timers:

```
switch# configure terminal
switch(config)# track 2 ip route 209.165.201.0/8 reachability
switch(config-track)# delay up 20 down 30
switch(config-track)# copy running-config startup-config
```

This example shows how to configure a track list with an up weight threshold of 30 and a down threshold of 10 with delay timers:

```
switch# configure terminal
switch(config)# track 1 list threshold weight
switch(config-track)# threshold weight up 30 down 10
switch(config-track)# object 10 weight 15
switch(config-track)# object 20 weight 15
switch(config-track)# object 30
switch(config-track)# delay up 20 down 30
```

This example shows the delay timer in the **show track** command output before and after an interface is shut down:

```
switch(config-track)# show track
Track 1
  Interface loopback1 Line Protocol
  Line Protocol is UP
  1 changes, last change 00:00:13
  Delay down 10 secs

switch(config-track)# interface loopback 1
switch(config-if)# shutdown
switch(config-if)# show track
Track 1
  Interface loopback1 Line Protocol
  Line Protocol is delayed DOWN (8 secs remaining) <----- delay timer counting down
  1 changes, last change 00:00:22
  Delay down 10 secs
```

Configuring Object Tracking for a Nondefault VRF

You can configure Cisco NX-OS to track an object in a specific VRF.

SUMMARY STEPS

1. **configure terminal**
2. **track *object-id* ip route *prefix/length* reachability**
3. **vrf member *vrf-name***
4. (Optional) **show track [*object-id*]**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	track <i>object-id</i> ip route <i>prefix/length</i> reachability Example: switch(config)# track 2 ip route 192.0.2.0/8 reachability switch(config-track)#	Creates a tracked object for a route and enters tracking configuration mode. The <i>object-id</i> range is from 1 to 500. The prefix format for IP is A.B.C.D/length, where the length range is from 1 to 32.
Step 3	vrf member <i>vrf-name</i> Example: switch(config-track)# vrf member Red	Configures the VRF to use for tracking the configured object.
Step 4	show track [<i>object-id</i>] Example: switch(config-track)# show track 3	(Optional) Displays object tracking information.
Step 5	copy running-config startup-config Example: switch(config-track)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure object tracking for a route and use VRF Red to look up reachability information for this object:

```
switch# configure terminal
switch(config)# track 2 ip route 209.165.201.0/8 reachability
switch(config-track)# vrf member Red
switch(config-track)# copy running-config startup-config
```

This example shows how to modify tracked object 2 to use VRF Blue instead of VRF Red to look up reachability information for this object:

```
switch# configure terminal
switch(config)# track 2
```

```
switch(config-track)# vrf member Blue
switch(config-track)# copy running-config startup-config
```

Verifying the Object Tracking Configuration

To display the object tracking configuration information, perform one of the following tasks:

Command	Purpose
<code>show track [object-id] [brief]</code>	Displays the object tracking information for one or more objects.
<code>show track [object-id] interface [brief]</code>	Displays the interface-based object tracking information.
<code>show track [object-id] ip route [brief]</code>	Displays the IPv4 route-based object tracking information.

Configuration Examples for Object Tracking

This example shows how to configure object tracking for route reachability and use VRF Red to look up reachability information for this route:

```
switch# configure terminal
switch(config)# track 2 ip route 209.165.201.0/8 reachability
switch(config-track)# vrf member Red
switch(config-track)# copy running-config startup-config
```

Related Topics

See the following topics for information related to object tracking:

- [Chapter 14, “Configuring Layer 3 Virtualization”](#)
- [Chapter 20, “Configuring HSRP”](#)

Additional References

For additional information related to implementing object tracking, see the following sections:

- [Related Documents, page 22-14](#)
- [Standards, page 22-14](#)

Related Documents

Related Topic	Document Title
Object Tracking CLI commands	<i>Cisco Nexus 3000 Series Command Reference</i>
Configuring the Embedded Event Manager	<i>Cisco Nexus 3000 Series System Management Configuration Guide</i>

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—



IETF RFCs

This appendix lists the IETF RFCs supported in Cisco NX-OS Release 6.x.

BGP RFCs

RFCs	Title
RFC 1997	<i>BGP Communities Attribute</i>
RFC 2385	<i>Protection of BGP Sessions via the TCP MD5 Signature Option</i>
RFC 2439	<i>BGP Route Flap Damping</i>
RFC 2519	<i>A Framework for Inter-Domain Route Aggregation</i>
RFC 2858	<i>Multiprotocol Extensions for BGP-4</i>
RFC 3065	<i>Autonomous System Confederations for BGP</i>
RFC 3392	<i>Capabilities Advertisement with BGP-4</i>
RFC 4271	<i>A Border Gateway Protocol 4 (BGP-4)</i>
RFC 4273	<i>Definitions of Managed Objects for BGP-4</i>
RFC 4456	<i>BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)</i>
RFC 4486	<i>Subcodes for BGP Cease Notification Message</i>
RFC 4893	<i>BGP Support for Four-octet AS Number Space</i>
RFC 5004	<i>Avoid BGP Best Path Transitions from One External to Another</i>
draft-ietf-idr-bgp4-mib-15.txt	<i>BGP4-MIB</i>

First-Hop Redundancy Protocols RFCs

RFCs	Title
RFC 2281	<i>Hot Standby Redundancy Protocol</i>
RFC 3768	<i>Virtual Router Redundancy Protocol</i>

IP Services RFCs

RFCs	Title
RFC 786	<i>UDP</i>
RFC 791	<i>IP</i>
RFC 792	<i>ICMP</i>
RFC 793	<i>TCP</i>
RFC 826	<i>ARP</i>
RFC 1027	<i>Proxy ARP</i>
RFC 1591	<i>DNS Client</i>
RFC 1812	<i>IPv4 routers</i>

OSPF RFCs

RFCs	Title
RFC 2328	<i>OSPF Version 2</i>
RFC 3101	<i>The OSPF Not-So-Stubby Area (NSSA) Option</i>
RFC 2370	<i>The OSPF Opaque LSA Option</i>
RFC 3137	<i>OSPF Stub Router Advertisement</i>

RIP RFCs

RFCs	Title
RFC 2453	<i>RIP Version 2</i>
RFC 2082	<i>RIP-2 MD5 Authentication</i>



A

ABR	See area border router .
address family	A specific type of network addressing supported by a routing protocol. Examples include IPv4 unicast.
adjacency	Two OSPF routers that have compatible configurations and have synchronized their link-state databases.
administrative distance	A rating of the trustworthiness of a routing information source. In general, the higher the value, the lower the trust rating.
area	A logical division of routers and links within an OSPF domain that creates separate subdomains. LSA flooding is contained within an area.
area border router	A router that connects one OSPF area to another OSPF area.
ARP	Address Resolution Protocol. ARP discovers the MAC address for a known IPv4 address.
AS	See autonomous system .
ASBR	See autonomous system border router .
attributes	Properties of a route that are sent in BGP UPDATE messages. These attributes include the path to the advertised destination as well as configurable options that modify the best path selection process.
autonomous system	A network controlled by a single technical administration entity.
autonomous system border router	A router that connect a an OSPF autonomous system to an external autonomous system.
AVF	Active virtual forwarder. A gateway within a GLBP group elected to forward traffic for a specified virtual MAC address.
AVG	Active virtual gateway. One virtual gateway within a GLBP group is elected as the active virtual gateway and is responsible for the operation of the protocol.

B

backup designated router	See BDR .
---------------------------------	---------------------------

bandwidth	The available traffic capacity of a link.
BDR	Backup designated router. An elected router in a multi-access OSPF network that acts as the backup if the designated router fails. All neighbors form adjacencies with the backup designated router (BDR) as well as the designated router.
BGP	Border Gateway Protocol. BGP is an interdomain or exterior gateway protocol.
BGP peer	A remote BGP speaker that is an established neighbor of the local BGP speaker.
BGP speaker	BGP-enabled router.

C

communication cost	Measure of the operating cost to route over a link.
converged	The point at which all routers in a network have identical routing information.
convergence	See converged .

D

dead interval	The time within which an OSPF router must receive a Hello packet from an OSPF neighbor. The dead interval is usually a multiple of the hello interval. If no Hello packet is received, the neighbor adjacency is removed.
default gateway	A router to which all unroutable packets are sent. Also called the router of last resort.
delay	The length of time required to move a packet from the source to the destination through the internetwork.
designated router	See DR .
DHCP	Dynamic Host Control Protocol.
Diffusing Update Algorithm	See DUAL .
distance vector	Defines routes by distance (for example, the number of hops to the destination) and direction (for example, the next-hop router) and then broadcasts to the directly connected neighbor routers.
DNS client	Domain Name System client. Communicates with DNS server to translate a host name to an IP address.
DR	Designated router. An elected router in a multi-access OSPF network that sends LSAs on behalf of all its adjacent neighbors. All neighbors establish adjacency with only the designated router and the backup designated router.
DUAL	Diffusing Update Algorithm. EIGRP algorithm used to select optimal routes to a destination.

E

eBGP	External Border Gateway Protocol (BGP). Operates between external systems.
EIGRP	Enhanced Interior Gateway Protocol. A Cisco routing protocol that uses the Diffusing Update Algorithm to provide fast convergence and minimized bandwidth utilization.

F

feasible distance	The lowest calculated distance to a network destination in EIGRP. The feasibility distance is the sum of the advertised distance from a neighbor plus the cost of the link to that neighbor.
feasible successor	Neighbors in EIGRP that advertise a shorter distance to the destination than the current feasibility distance.
FIB	Forwarding Information Base. The forwarding table on each module that is used to make the Layer 3 forwarding decisions per packet.

G

gateway	A switch or router that forwards Layer 3 traffic from a LAN to the rest of the network.
----------------	---

H

hello interval	The configurable time between each hello packet sent by an OSPF or EIGRP router.
hello packet	A special message used by OSPF or IS-IS to discover neighbors. Also acts as a keepalive message between established neighbors.
hold time	<p>In BGP, the maximum time limit allowed in BGP between UPDATE or KEEPALIVE messages. If this time is exceeded, the TCP connection between the BGP peers is closed.</p> <p>In EIGRP, the maximum time allowed between EIGRP hello messages. If this time is exceeded, the neighbor is declared unreachable.</p>
hop count	The number of routers that can be traversed in a route. Used by RIP.

I

iBGP	Internal Border gateway protocol (BGP). Operates within an autonomous system.
ICMP	
IETF RFCs	Internet Engineering Task Force Request for Comments.

IGP	Interior gateway protocol. Used between routers within the same autonomous system.
instance	An independent, configurable entity, typically a protocol.
IP tunnels	
IPv4	Internet Protocol version 4.
<hr/>	
K	
keepalive	A special message sent between routing peers to verify and maintain communications between the pair.
<hr/>	
L	
link cost	An arbitrary number configured on an OSPF interface which is in shortest path first calculations.
link-state	Shares information about a link, link cost to neighboring routers.
link-state advertisement	See LSA .
LSA	Link-state advertisement. An OSPF message to share information on the operational state of a link, link cost, and other OSPF neighbor information.
link-state database	OSPF database of all LSAs received. OSPF uses this database to calculate the best path to each destination in the network.
link-state refresh	The time that OSPF floods the network with LSAs to ensure all OSPF routers have the same information.
load	The degree to which a network resource, such as a router, is busy.
load balancing	The distribution of network traffic across multiple paths to a given destination.
<hr/>	
M	
message digest	A one-way hash applied to a message using a shared password and appended to the message to authenticate the message and ensure the message has not been altered in transit.
metric	A standard of measurement, such as the path bandwidth, that is used by routing algorithms to determine the optimal path to a destination.
MD5 authentication digest	A cryptographic construction that is calculated based on an authentication key and the original message and sent along with the message to the destination. Allows the destination to determine the authenticity of the sender and guarantees that the message has not been tampered with during transmission.
MTU	Maximum transmission unit. The largest packet size that a network link will transmit without fragmentation.

N

network layer reachability information	BGP network layer reachability information (NRLI). Contains the a list of network IP addresses and network masks for networks that are reachable from the advertising BGP peer.
next-hop	The next router that a packet is sent to on its way to the destination address.
NSSA	Not-So-Stubby-Area. Limits AS external LSAs in an OSPF area.

O

OSPF	Open Shortest Path First. An IETF link-state protocol. OSPFv2 supports IPv4.
-------------	--

P

path length	Sum of all link costs or the hop count that a packet experiences when routed from the source to the destination.
--------------------	--

R

redistribution	One routing protocol accepts route information from another routing protocol and advertises it in the local autonomous system.
Reliable Transport Protocol	Responsible for guaranteed, ordered delivery of EIGRP packets to all neighbors.
reliability	The dependability (usually described in terms of the bit-error rate) of each network link.
RIB	Routing Information Base. Maintains the routing table with directly connected routes, static routes, and routes learned from dynamic unicast routing protocols.
routing information base	See RIB .
route map	A construct used to map a route or packet based on match criteria and optionally alter the route or packet based on set criteria. Used in route redistribution.
route summarization	A process that replaces a series of related, specific routes in a route table with a more generic route.
router ID	A unique identifier used by routing protocols. If not manually configured, the routing protocol selects the highest IP address configured on the system.

S

SPF algorithm	Shortest Path First algorithm. Dijkstra's algorithm used by OSPF to determine the shortest route through a network to a particular destination.
split horizon	Routes learned from an interface are not advertised back along the interface they were learned on, preventing the router from seeing its own route updates.
split horizon with poison reverse	Routes learned from an interface are set as unreachable and advertised back along the interface they were learned on, preventing the router from seeing its own route updates.
static route	A manually configured route.
stub area	An OSPF area that does not allow AS External (type 5) LSAs.
stub router	A router that has no direct connection to the main network and which routes to that network using a known remote router.
SVI	Switched Virtual Interface.

U

UFIB	Unicast IPv4 forwarding information base.
URIB	Unicast IPv4 routing information base. The unicast routing table that gathers information from all routing protocols and updates the forwarding information base for each module.

V

virtualization	A method of making a physical entity act as multiple, independent logical entities.
VRF	Virtual Routing and Forwarding. A method used to create separate, independent Layer 3 entities within a system.
VRRP	Virtual Router Redundancy Protocol.



A

ABR [4-4](#)

address formats

IPv4 [2-2](#)

IPv6 [3-2](#)

IPv6 (table) [3-2](#)

address resolution protocol. See ARP

administrative distance

description [1-7](#)

static routing [13-2](#)

aggregatable global addresses. See IPv6

unicast addresses

areas [5-5](#)

ARP

caching [2-3](#)

configuring gratuitous ARP [2-13](#)

configuring Local Proxy ARP [2-12](#)

configuring Proxy ARP [2-11](#)

configuring static ARP entries [2-10](#)

description [2-3](#)

gratuitous ARP [2-5](#)

Local Proxy ARP [2-5](#)

process (figure) [2-3](#)

Proxy ARP [2-5](#)

Reverse ARP [2-4](#)

AS. See autonomous system

ASBR [4-5](#)

AS confederations

configuring [8-29](#)

description [8-5](#)

AS numbers

4-byte support. [1-5](#)

ranges (table) [1-5](#)

AS-path lists

configuring [16-8](#)

description [16-4](#)

autonomous system

description [1-5](#)

B

bandwidth [1-4](#)

BDR [4-3](#)

BFD

BGP [8-9, 13-3](#)

configuring for BGP [18-12, 18-14, 18-18](#)

configuring on an interface [18-8, 18-9](#)

configuring session parameters [18-6](#)

description [18-1](#)

echo function [18-2](#)

enable the feature [18-6](#)

licensing [18-3](#)

monitoring [18-23](#)

OSPF [4-11](#)

OSPFv3 [5-12](#)

session parameters [18-2](#)

verifying [18-23](#)

virtualization support [18-3](#)

BGP [7-7](#)

administrative distances (table) [7-2](#)

BFD [8-9, 13-3](#)

clearing neighbors [7-22](#)

conditional advertisement [8-7](#)

conditional advertisement example [8-38](#)

configuration modes [7-8](#)

- configuring conditional advertisement [8-36](#)
- configuring dynamic capability [8-34](#)
- configuring maximum prefixes [8-34](#)
- configuring prefix peering [8-21](#)
- configuring route dampening [8-33](#)
- default settings [7-10, 8-12](#)
- description [7-1, 7-7, 8-1](#)
- disable the feature [7-12](#)
- displaying statistics [7-27, 8-49](#)
- eBGP [8-3](#)
- enable the feature [7-11](#)
- example configuration [7-27](#)
- generic specific extended community lists [16-5](#)
- guidelines [7-8, 8-11](#)
- iBGP [8-4](#)
- licensing requirements [7-7, 8-10](#)
- limitations [7-8, 8-11](#)
- MIBs [6-28, 7-28](#)
- modifying AS-paths [8-43](#)
- modifying next-hop address [8-23](#)
- MP-BGP [8-10](#)
- next-hop address tracking [8-8](#)
- path selection [7-4](#)
- prerequisites [7-7, 8-10](#)
- router ID [7-3](#)
- speakers [7-1](#)
- tuning [8-42](#)
- unicast RIB [7-7](#)
- verifying configuration [7-25, 8-48](#)
- virtualization support [7-7](#)
- BGP aggregate addresses
 - configuring [8-35](#)
- BGP AS-path lists
 - configuring [16-8](#)
 - description [16-4](#)
- BGP authentication
 - configuring [8-22](#)
 - description [8-2](#)
- BGP autonomous systems
 - description [7-2](#)
- BGP capabilities negotiation
 - description [8-6](#)
 - disabling [8-25](#)
- BGP community lists
 - configuring [16-9, 16-11](#)
 - description [16-4](#)
- BGP extended community lists
 - description [16-4](#)
- BGP instance
 - creating [7-12](#)
 - deleting [7-13](#)
 - restarting [7-13](#)
- BGP load balancing
 - configuring [8-34](#)
- BGP loadsharing
 - description [8-7](#)
- BGP multipath. See BGP loadsharing
- BGP peers
 - authentication (note) [8-2](#)
 - configuring [7-14, 7-17, 7-20](#)
 - description [7-3](#)
- BGP route aggregation
 - description [8-7](#)
- BGP route dampening [8-6](#)
- BGP route redistribution
 - configuring [8-38](#)
 - description [8-9](#)
- BGP sessions
 - reset options [8-3](#)
 - resetting [8-22](#)
 - route policies [8-3](#)
- BGP templates
 - configuring peer-policy templates [8-16](#)
 - configuring peer templates [8-18](#)
 - configuring session templates [8-13](#)
 - description [8-2](#)
 - peer-policy templates [8-2](#)
 - peer-session templates [8-2](#)

peer template [8-2](#)

Border Gateway Protocol. See BGP

C

CDP [3-11](#)

communication cost [1-4](#)

community lists

configuring [16-9, 16-11](#)

description [16-4](#)

comparing

link-state and distance vector routing algorithms [1-9](#)

D

default gateway

description [1-8](#)

default settings

BGP [7-10, 8-12](#)

ECMP for host routes [10-1, 11-2](#)

EIGRP [6-8](#)

HSRP [20-6](#)

IP [2-7](#)

IPv6 [3-17](#)

object tracking [22-3](#)

OSPF [4-12](#)

OSPFv3 [5-13](#)

policy-based routing [17-4](#)

RIP [12-4](#)

Route Policy Manager [16-6](#)

static routing [13-4](#)

VRF [14-6](#)

VRRP [21-6](#)

delay [1-4](#)

distance vector routing algorithms [1-9](#)

distribution

RIP [12-3](#)

DNS [3-10](#)

DR [4-3](#)

E

eBGP

configuring [8-25](#)

configuring AS confederations [8-29](#)

configuring multihop [8-27](#)

description [8-3](#)

disabling fast external failover [8-28](#)

disabling single-hop checking [8-26, 8-27](#)

limiting the AS-path attribute [8-28](#)

eBGP AS confederations. See AS confederations

ECMP. See equal cost multipath

ECMP for host routes

default settings [10-1, 11-2](#)

example configuration [10-3, 11-4](#)

information about [11-1](#)

licensing requirements [11-2](#)

verifying configuration [11-4](#)

EIGRP

authentication [6-5](#)

configuring a summary address [6-16](#)

configuring authentication [6-13](#)

configuring hello interval [6-22](#)

configuring load balancing [6-21](#)

configuring route redistribution [6-17](#)

configuring stub routing [6-16](#)

creating an instance [6-10](#)

default settings [6-8](#)

deleting an instance [6-11](#)

description [6-1](#)

disabling an instance [6-12](#)

disabling split horizon [6-23](#)

disabling the feature [6-9](#)

displaying statistics [6-27](#)

DUAL algorithm [6-2](#)

ECMP [6-6](#)

enabling the feature [6-9](#)

- example configuration [6-28](#)
- external route metrics [6-4](#)
- guidelines [6-7](#)
- hold time [6-2](#)
- internal route metrics [6-3](#)
- licensing requirements [6-7](#)
- limitations [6-7](#)
- limit redistributed routes [6-19](#)
- load balancing [6-6](#)
- neighbor discovery [6-2](#)
- prerequisites [6-7](#)
- restarting an instance [6-12](#)
- route redistribution [6-6](#)
- route summarization [6-6](#)
- route updates [6-3](#)
- shutting down on an interface [6-12, 6-13](#)
- split horizon [6-6](#)
- stub routers [6-5](#)
- tuning [6-23](#)
- unicast RIB [6-4](#)
- verifying configuration [6-27](#)
- virtualization support [6-7](#)

eigrp

- passive interface [6-12](#)

- equal cost multipath [1-6](#)

- extended community lists

- description [16-4](#)

- external BGP. See eBGP

F

FIB

- clearing routes [15-9](#)
- description [1-12, 15-1](#)
- displaying [15-3](#)
- licensing requirements [15-3](#)
- verifying [15-10](#)
- virtualization support [15-3](#)
- VRFs [1-12](#)

Fibre Channel interfaces

- default settings [18-4](#)

forwarding

- adjacency manager [1-11](#)

- architecture [1-10, 15-1](#)

- FIB [1-11](#)

- unicast forwarding distribution module [1-11](#)

forwarding information base. See FIB

G

- glean throttling [2-5](#)

- maximum number [2-16](#)

- timeout [2-17](#)

graceful restart

- configuring in OSPFv3 [5-36](#)

gratuitous ARP

- configuring [2-13](#)

- description [2-5](#)

H

Hot Standby Router Protocol. See HSRP

HSRP

- addressing [20-3](#)

- configuring a group [20-8](#)

- configuring priority [20-14](#)

- customizing [20-14](#)

- default settings [20-6](#)

- description [20-2, 20-5](#)

- disabling the feature [20-7](#)

- enabling the feature [20-7](#)

- example configuration [20-16](#)

- guidelines [20-6](#)

- licensing requirements [20-6](#)

- limitations [20-6](#)

- load sharing [20-4](#)

- messages [20-4](#)

- prerequisites [20-6](#)
 - standby router [20-2](#)
 - verifying configuration [20-16](#)
 - virtualization support [20-5](#)
 - HSRP authentication
 - configuring [20-10](#)
 - description [20-4](#)
 - HSRP object tracking
 - configuring [20-12](#)
 - description [20-5](#)
 - HSRP versions
 - configuring [20-8](#)
 - description [20-4](#)
 - HSRP virtual MAC address
 - configuring [20-10](#)
 - description [20-2](#)
-
- iBGP
 - configuring route reflector [8-31](#)
 - description [8-4](#)
 - iBGP route reflector. See route reflector
 - ICMP
 - description [2-6](#)
 - with local proxy ARP (note) [2-6](#)
 - ICMPv6 [3-11](#)
 - packet header format (figure) [3-11](#)
 - IDS, enabling [3-26](#)
 - information about
 - ECMP for host routes [11-1](#)
 - interfaces
 - default settings [18-4](#)
 - internal BGP. See iBGP
 - Internet Control Message Protocol. See ICMP
 - IP
 - addresses [2-2](#)
 - ARP. See ARP
 - configuring addresses [2-8](#)
 - configuring secondary addresses [2-9](#)
 - default settings [2-7](#)
 - description [2-1, 2-6](#)
 - example configuration [2-19](#)
 - guidelines [2-7](#)
 - ICMP. See ICMP
 - licensing requirements [2-7](#)
 - limitations [2-7](#)
 - packet header [3-7](#)
 - prerequisites [2-7](#)
 - secondary addresses (note) [2-2](#)
 - subnet masks [2-1](#)
 - verifying configuration [2-19](#)
 - virtualization support [2-6](#)
 - IPv4. See IP
 - IPv6 [7-3](#)
 - addresses compatible with IPv4 [3-5](#)
 - address formats [3-2](#)
 - address formats (table) [3-2](#)
 - CDP [3-11](#)
 - configuring addresses [3-17](#)
 - configuring neighbor discovery [3-23](#)
 - default settings [3-17](#)
 - description [3-1](#)
 - DNS [3-10](#)
 - enabling IDS checks [3-26](#)
 - enabling packet verification [3-26](#)
 - EUI-64 format [3-4](#)
 - example configuration [3-27](#)
 - guidelines [3-16](#)
 - ICMP [3-11](#)
 - interface ID [3-4](#)
 - licensing requirements [3-16](#)
 - limitations [3-16](#)
 - link-local addresses [3-5](#)
 - loopback address (note) [3-3](#)
 - neighbor discovery [3-12](#)
 - neighbor redirect message [3-15](#)
 - neighbor solicitation message [3-12](#)

- packet header [3-7](#)
- path MTU discovery [3-10](#)
- prerequisites [3-16](#)
- RFC [3-3, 3-4](#)
- router advertisement message [3-13](#)
- site-local address [3-7](#)
- subnet ID [3-4](#)
- unicast addresses [3-3](#)
- unique local addresses [3-6](#)
- unspecified address (note) [3-3](#)
- verifying configuration [3-27](#)
- virtualization support [3-16](#)

L

- Layer 3 consistency checker
 - description [15-2](#)
 - triggering [15-8](#)
- licensing requirements [7-7](#)
 - BGP [8-10](#)
 - ECMP for host routes [11-2](#)
 - EIGRP [6-7](#)
 - FIB [15-3](#)
 - HSRP [20-6](#)
 - IP [2-7](#)
 - IPv6 [3-16](#)
 - object tracking [22-3](#)
 - OSPF [4-12](#)
 - OSPFv3 [5-12](#)
 - policy-based routing [17-3](#)
 - RIP [12-4](#)
 - Route Policy Manager [16-5](#)
 - static routing [13-3](#)
 - uRIB [15-3](#)
 - VRF [14-5](#)
 - VRRP [21-5](#)
- link-state advertisements [4-1](#)
- link-state routing algorithms [1-9](#)
- load [1-4](#)

- load balancing [1-6](#)
- Local Proxy ARP
 - configuring [2-12](#)
 - description [2-5](#)
- LSAs [5-5](#)
 - for OSPFv3 (table) [5-6](#)

M

- MAC lists
 - description [16-2](#)
- management interfaces
 - default settings [18-4](#)
- mgmt0 interfaces
 - default settings [18-4](#)
- MIBs
 - BGP [6-28, 7-28](#)
 - OSPF [4-43, 20-17](#)
 - OSPFv3 [5-42](#)
- MP-BGP [8-10](#)
 - configuring [8-40](#)
- Multiprotocol BGP
 - see MP-BGP

N

- ND
 - configuring [3-23](#)
 - description [3-12](#)
- neighbor discovery. See ND
- neighbor redirect message [3-15](#)
- new and changed features (table) [1-1](#)
- next hop [1-2](#)
- NSSA [4-9](#)
 - configuring [4-26](#)
 - description for OSPFv3 [5-9](#)

O

object tracking

- configuring a delay [22-10](#)
- configuring a track list with boolean expression [22-6](#)
- configuring a track list with percentage [22-7, 22-8](#)
- configuring for a nonDefault VRF [22-12](#)
- configuring for route reachability [22-5](#)
- configuring on an interface [22-4](#)
- default settings [22-3](#)
- description [22-1](#)
- example configuration [22-13](#)
- guidelines [22-3](#)
- licensing requirements [22-3](#)
- limitations [22-3](#)
- track list [22-2](#)
- verifying configuration [22-13](#)
- virtualization support [22-2](#)

Open Shortest Path First. See OSPF

Open Shortest Path First version 3. See OSPFv3

OSPF

- adjacency [4-1, 4-3](#)
- area border router [4-4](#)
- areas [4-1, 4-4](#)
- AS border router [4-5](#)
- authentication [4-7](#)
- backup designated router [4-3](#)
- BFD [4-11](#)
- configuring area authentication [4-19](#)
- configuring a totally stubby area [4-25](#)
- configuring authentication [4-18](#)
- configuring authentication on an interface [4-20](#)
- configuring DR priority [4-18](#)
- configuring ECMP [4-16](#)
- configuring filter lists [4-23](#)
- configuring load balancing [4-16](#)
- configuring MD5 authentication [4-20](#)
- configuring networks [4-16](#)
- configuring NSSA [4-26](#)

- configuring on an interface [4-16](#)
- configuring optional parameters on an interface [4-17](#)
- configuring redistribution [4-30](#)
- configuring route summarization [4-34](#)
- configuring simple password authentication [4-20](#)
- configuring stub areas [4-24](#)
- configuring stub route advertisements [4-35](#)
- configuring the hello interval [4-18](#)
- configuring virtual links [4-28](#)
- creating an instance [4-14](#)
- dead interval [4-2](#)
- default settings [4-12](#)
- delete an instance [4-15](#)
- description [4-1](#)
- designated router [4-3](#)
- disable the feature [4-14](#)
- displaying statistics [4-42](#)
- enable the feature [4-13](#)
- example configuration [4-42](#)
- guidelines [4-12](#)
- hello interval [4-2](#)
- hello packet [4-2](#)
- licensing requirements [4-12](#)
- limitations [4-12](#)
- link cost [4-6](#)
- link-state database [4-7](#)
- LSA [4-1](#)
- LSA flooding [4-6](#)
- LSA pacing [4-6](#)
- LSAs [4-5, 4-7](#)
- LSA types (table) [4-6](#)
- MIBs [4-43, 20-17](#)
- modifying default timers [4-36](#)
- multiple instances [4-11](#)
- neighbors [4-2](#)
- not-so-stubby area [4-9](#)
- NSSA [4-9](#)
- opaque LSAs [4-7](#)
- prerequisites [4-12](#)

- redistributed routes [4-32](#)
- restarting an instance [4-39](#)
- route redistribution
 - description [4-10](#)
- route summarization
 - description [4-10](#)
- shutting down an instance [4-18](#)
- SPF optimization [4-11](#)
- stub area [4-8](#)
- stub area (figure) [4-9](#)
- stub router advertisements
 - description [4-11](#)
- unicast RIB [4-7](#)
- verifying configuration [4-41](#)
- virtualization support [4-12](#)
- virtual link [4-9](#)
- virtual link (figure) [4-10](#)

OSPFv2. See OSPF

OSPFv2 (Open Shortest Path First Version 2)

- description [5-1](#)

OSPFv3

- address families [5-8](#)
- adjacency [5-3](#)
- areas [5-5](#)
- BFD [5-12](#)
- comparison to OSPFv2 [5-2](#)
- configuring ECMP [5-17](#)
- configuring filter lists [5-20](#)
- configuring graceful restart [5-36](#)
- configuring load balancing [5-17](#)
- configuring networks [5-17](#)
- configuring NSSA [5-23](#)
- configuring redistribution [5-29](#)
- configuring route summarization [5-32](#)
- configuring stub areas [5-22](#)
- configuring totally stubby areas [5-23](#)
- configuring virtual links [5-27](#)
- configuring with VRFs [5-38](#)
- creating an instance [5-15](#)

- default settings [5-13](#)
- description [5-1](#)
- displaying statistics [5-41](#)
- enabling the feature [5-14](#)
- example configuration [5-41](#)
- guidelines [5-13](#)
- licensing requirements [5-12](#)
- limitations [5-13](#)
- link cost [5-6](#)
- link-state database [5-7](#)
- LSA flooding [5-7](#)
- LSA pacing [5-7](#)
- LSAs [5-5](#)
- LSA types (table) [5-6](#)
- MIBs [5-42](#)
- modifying default timers [5-34](#)
- multiple instances [5-12](#)
- neighbors [5-3](#)
- NSSA [5-9](#)
- prerequisites [5-13](#)
- redistributed routes [5-30](#)
- restarting an instance [5-37](#)
- RFC [5-2](#)
- route redistribution [5-10](#)
- route summarization [5-10](#)
- SPF optimization [5-12](#)
- unicast RIB [5-8](#)
- verifying configuration [5-40](#)
- virtualization support [5-12](#)
- virtual links [5-10](#)

P

- path length [1-4](#)
- path MTU discovery [3-10](#)
- policy-based routing
 - configuring a route policy [17-5](#)
 - configuring match parameters [17-7](#)
 - configuring set parameters [17-7](#)

- default settings [17-4](#)
- description [17-1](#)
- disabling [17-5](#)
- enabling [17-4](#)
- example configuration [17-8](#)
- feature history (table) [17-10](#)
- guidelines [17-3](#)
- licensing requirements [17-3](#)
- limitations [17-3](#)
- prerequisites [17-3](#)
- route maps [17-2](#)
- set criteria [17-2](#)
- verifying configuration [17-8](#)
- policy route maps
 - description [17-2](#)
- prefix lists
 - configuring [16-6](#)
 - description [16-2](#)
- Proxy ARP
 - configuring [2-11](#)
 - description [2-5](#)

R

- redistribution
 - description [1-6](#)
- redistribution [1-5](#)
 - BGP [8-9](#)
 - configuring for OSPF [4-30](#)
 - configuring for OSPFv3 [5-29](#)
 - configuring in BGP [8-38](#)
 - configuring in RIP [12-12](#)
 - configuring on EIGRP [6-17](#)
 - EIGRP [6-6](#)
 - maximum limit for EIGRP [6-19](#)
 - maximum limit for OSPF [4-32](#)
 - maximum limit for OSPFv3 [5-30](#)
 - with route maps [16-5](#)
- reliability [1-4](#)

- Reverse ARP
 - description [2-4](#)
 - limitations [2-4](#)
 - RFC [2-4](#)
- RIB
 - description [1-11, 15-2](#)
 - see uRIB
- RIP
 - clearing statistics [12-17](#)
 - configuring a passive interface [12-11](#)
 - configuring on an interface [12-8](#)
 - default settings [12-4](#)
 - description [12-2](#)
 - disable the feature [12-6](#)
 - displaying statistics [12-17](#)
 - enabling the feature [12-5](#)
 - example configuration [12-18](#)
 - guidelines [12-4](#)
 - licensing requirements [12-4](#)
 - limitations [12-4](#)
 - prerequisites [12-4](#)
 - route filtering [12-3](#)
 - tuning [12-16](#)
 - verifying configuration [12-17](#)
 - virtualization support [12-4](#)
- RIP authentication
 - configuring [12-9](#)
 - description [12-2](#)
- RIP instance
 - creating [12-6](#)
 - deleting [12-7](#)
 - optional parameters [12-7](#)
 - restarting [12-8](#)
- RIP load balancing
 - configuring [12-7](#)
 - description [12-3](#)
- RIP route distribution
 - description [12-3](#)
- RIP route redistribution

- configuring [12-12](#)
 - RIP route summarization
 - configuring [12-11](#)
 - description [12-3](#)
 - RIP split horizon
 - configuring with poison reverse [12-11](#)
 - description [12-2](#)
 - route maps
 - configuring [16-12](#)
 - configuring match parameters [16-13](#)
 - configuring set parameters [16-14](#)
 - description [16-3](#)
 - example configuration [16-17](#)
 - for policy -based routing [17-2](#)
 - match criteria [16-3](#)
 - redistribution [16-5](#)
 - set changes [16-3](#)
 - route metric
 - bandwidth [1-4](#)
 - communication cost [1-4](#)
 - delay [1-4](#)
 - load [1-4](#)
 - path length [1-4](#)
 - reliability [1-4](#)
 - route policy
 - configuring [17-5](#)
 - configuring match parameters [17-7](#)
 - configuring set parameters [17-7](#)
 - description [17-1](#)
 - example configuration [17-8](#)
 - Route Policy Manager
 - default settings [16-6](#)
 - example configuration [16-17](#)
 - guidelines [16-5](#)
 - licensing requirements [16-5](#)
 - limitations [16-5](#)
 - route policy manager
 - description [16-1](#)
 - verifying configuration [16-17](#)
 - router advertisement message [3-13](#)
 - route redistribution
 - OSPFv3 [5-10](#)
 - route reflector
 - configuring [8-31](#)
 - description [8-5](#)
 - router ID
 - description [1-5](#)
 - routes, estimating memory requirements [15-9](#)
 - route summarization
 - configuring [4-34](#)
 - configuring on EIGRP [6-16](#)
 - EIGRP [6-6](#)
 - OSPFv3 [5-10, 5-32](#)
 - RIP [12-3](#)
 - route table
 - description [1-2](#)
 - routing algorithms
 - distance vector [1-9](#)
 - link-state [1-9](#)
 - Routing Information Protocol. See RIP
 - routing metrics
 - description [1-2](#)
 - routing protocols
 - administrative distance [1-7](#)
 - comparing link-state algorithms to distance vector algorithms [1-9](#)
 - convergence, convergence [1-6](#)
 - description [1-1, 1-8](#)
 - distance vector [1-9](#)
 - link-state [1-9](#)
 - next hop [1-2](#)
 - redistribution [1-5, 1-6](#)
 - virtualization [1-10](#)
-
- ## S
- static routes
 - description [1-8](#)

- virtualization support [13-3](#)
- with ARP [2-3](#)
- static routing
 - administrative distance [13-2](#)
 - configuring [13-4](#)
 - configuring with VRFs [13-5](#)
 - default settings [13-4](#)
 - description [13-1](#)
 - example configuration [13-6](#)
 - guidelines [13-3](#)
 - licensing requirements [13-3](#)
 - limitations [13-3](#)
 - prerequisites [13-3](#)
 - verifying configuration [13-6](#)
- stub routing
 - description [1-7](#)

U

- uRIB
 - clearing routes [15-10](#)
 - description [15-1](#)
 - displaying [15-7](#)
 - displaying (example) [15-8](#)
 - Layer 3 consistency checker [15-2](#)
 - licensing requirements [15-3](#)
 - verifying [15-10](#)
 - virtualization support [15-3](#)

V

- virtualization
 - description [1-10](#)
- Virtual Router Redundancy Protocol. See VRRP
- VRF
 - assigning an interface to a VRF [14-8](#)
 - configuring routing parameters [14-9](#)
 - creating [14-7](#)

- default settings [14-6](#)
 - deleting [14-8](#)
 - example configuration [14-19](#)
 - guidelines [14-5](#)
 - licensing requirements [14-5](#)
 - limitations [14-5](#)
 - setting the routing context [14-12](#)
 - setting the scope [14-12](#)
 - verifying configuration [14-19](#)
- VRF-aware services
 - configuring [14-11](#)
 - description [14-3](#)
- VRF filtering
 - description [14-4](#)
 - example configuration [14-12](#)
- VRF-Lite
 - description [14-2](#)
 - guidelines [14-5](#)
 - limitations [14-5](#)
- VRF reachability
 - description [14-3](#)
 - example configuration [14-12](#)
- VRRP
 - benefits [21-3](#)
 - configuring time intervals for advertisement packets [21-12](#)
 - default settings [21-6](#)
 - description [21-1, 21-5](#)
 - disabling the feature [21-7](#)
 - displaying statistics [21-22](#)
 - enabling the feature [21-7, 21-17](#)
 - example configuration [21-22](#)
 - guidelines [21-6](#)
 - licensing requirements [21-5](#)
 - limitations [21-6](#)
 - verifying configuration [21-21](#)
 - virtualization support [21-5](#)
- VRRP advertisements
 - description [21-5](#)

VRRP authentication

configuring [21-10](#)description [21-5](#)

VRRP groups

configuring [21-7, 21-17](#)description [21-3](#)

VRRP priority

configuring [21-9](#)description [21-4](#)disabling preemption [21-13](#)preemption [21-4](#)

VRRP tracking

configuring [21-15](#)