Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide, Release 10.5(x)

First Published: 2024-07-26

Last Modified: 2024-11-27

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE. EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at: https://www.cisco.com/c/en/us/about/legal/cloud-and-software/software-terms.html. Cisco product warranty information is available at https://www.cisco.com/c/en/us/products/warranty-listing.html. US Federal Communications Commission Notices are found here https://www.cisco.com/c/en/us/products/us-fcc-notice.html.

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

Any products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

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

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

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

© 2024 Cisco Systems, Inc. All rights reserved.

CONTENTS

Trademarks ?

PREFACE

はじめに xxxiii

対象読者 xxxiii

表記法 xxxiii

Cisco Nexus 9000 シリーズ スイッチの関連資料 xxxiv

マニュアルに関するフィードバック xxxiv

通信、サービス、およびその他の情報 xxxv

Cisco バグ検索ツール xxxv

マニュアルに関するフィードバック xxxv

CHAPTER 1

New and Changed Information 1

New and Changed Information 1

CHAPTER 2

概要 3

ライセンス要件 3

サポートされるプラットフォーム 3

レイヤ 3 ユニキャスト ルーティングについて 3

ルーティングの基礎 4

パケット交換 4

ルーティング メトリック 5

パス長 5

Reliability 5

ルーティング遅延 5

帯域幅 5

負荷 5
通信コスト 6
ルータ ID 6
自律システム 6
コンバージェンス 7
ロード バランシングおよび等コスト マルチパス 7
ルートの再配布の概要 8
アドミニストレーティブ ディスタンス 8
スタブ ルーティング 8
ルーティング アルゴリズム 10
スタティック ルートおよびダイナミック ルーティング プロトコル 10
内部および外部ゲートウェイ プロトコル 10
ディスタンス ベクトル プロトコル 11
リンクステート プロトコル 11
レイヤ 3 仮想化 12
Cisco NX-OS フォワーディング アーキテクチャ 12
ユニキャスト RIB 13
隣接マネージャ 13
ユニキャスト転送分散モジュール 13
FIB 14
ハードウェア フォワーディング 14
ソフトウェア転送 14
レイヤ 3 ユニキャスト ルーティング機能のまとめ 15
IPv4 and IPv6 15
IP サービス 15
Open Shortest Path First（OSPF） 15
EIGRP 15
IS-IS 15
BGP 16
RIP 16
スタティック ルーティング 16
レイヤ 3 仮想化 16

Route Policy Manager 17
ポリシーベース ルーティング 17
ファーストホップ冗長プロトコル（FHRP） 17
オブジェクト トラッキング 17
関連項目 18

CHAPTER 3

IPv4 の設定 19
IPv4 の概要 19
複数の IPv4 アドレス 20
LPMルーティングモード 20
ホストから LPM へのスピルオーバー 23
アドレス解決プロトコル 23
ARP キャッシング 24
ARP キャッシュのスタティックおよびダイナミック エントリ 24
ARP を使用しないデバイス 25
Reverse ARP 25
プロシキ ARP 26
ローカル プロキシ ARP 26
Gratuitous ARP 26
MAC 削除時の定期的な ARP 更新 26
収集スロットル 27
パス MTU ディスカバリ 27
ICMP 27
IPv4 の仮想化のサポート 28
IPv4の前提条件 28
IPv4 の注意事項および制約事項 28
デフォルト設定 30
IPv4 の設定 31
IPv4 アドレス指定の設定 31
複数の IP アドレスの設定 32
最大ホスト ルーティング モードの設定 33

非階層ルーティング モードの設定（Cisco Nexus 9500 プラットフォーム スイッチのみ） 35
64 ビット ALPM ルーティング モードの設定（Cisco Nexus 9500 プラットフォーム スイッチのみ） 36
ALPM ルーティング モードの設定（Cisco Nexus 9300 プラットフォーム スイッチのみ） 37
LPM ヘビー ルーティング モードの設定（Cisco Nexus 9200 および 9300-EX プラットフォーム スイッチおよび 9732C-EX ライン カードのみ） 38
LPM インターネット ピアリング ルーティング モードの設定 40
LPM デュアルホスト ルーティング モードの構成 41
スタティック ARP エントリの設定 42
プロキシ ARP の設定 43
イーサネット インターフェイスでのローカル プロキシ ARP の設定 44
SVI でのローカル プロキシ ARP の設定 45
SVI の MAC 削除での定期的な ARP リフレッシュの構成 46
無償 ARP の設定 47
サブネット外の ARP 解決の構成 48
SVI インターフェイスごとの ARP キャッシュの構成 49
パス MTU ディスカバリの設定 50
IP ダイレクト ブロードキャストの設定 51
IP 収集スロットルの設定 52
ハードウェア IP 収集スロットルの最大値の設定 53
ハードウェア IP 収集スロットルのタイムアウトの設定 53
ICMP 送信元 IP フィールドのインターフェイス IP アドレスの設定 54
IPv4 リダイレクト Syslog の構成 55
IPv4 設定の確認 56
その他の参考資料 57
IPv4 の関連資料 57

CHAPTER 4
IPv6 の設定 59
IPv6 について 59
IPv6 アドレス形式 60

IPv6 ユニキャスト アドレス 61
集約可能グローバル アドレス 61
リンクローカル アドレス 63
IPv4 互換 IPv6 アドレス 63
ユニーク ローカル アドレス 63
サイト ローカル アドレス 64
IPv4 パケット ヘッダー 64
簡易 IPv6 パケット ヘッダー 65
IPv6 の DNS 68
IPv6 のパス MTU ディスカバリ 69
CDP IPv6 アドレスのサポート 69
IPv6 の ICMP 69
IPv6 ネイバー探索 70
IPv6 ネイバー送信要求メッセージ 71
IPv6 ステートレス自動設定 72
IPv6 コンピューティング ノード IP 自動構成 73
IPv6 ルータ アドバタイズメント メッセージ 73
IPv6 ネイバー リダイレクト メッセージ 74
IPv6 エニーキャスト アドレス 75
IPv6 マルチキャスト アドレス 76
LPMルーティングモード 77
ホストから LPM へのスピルオーバー 79
仮想化のサポート 80
ECMP を使用した IPv6 ルート 80
IPv6の前提条件 80
IPv6 の注意事項および制約事項 80
IPv6 の設定 82
IPv6 アドレッシングの設定 82
最大ホスト ルーティング モードの設定（Cisco Nexus 9500 プラットフォーム スイッチのみ） 84
非階層ルーティング モードの設定（Cisco Nexus 9500 シリーズ スイッチのみ） 85

64 ビット ALPM ルーティング モードの設定（Cisco Nexus 9500 プラットフォーム スイッチのみ） 87
ALPM ルーティング モードの設定（Cisco Nexus 9300 プラットフォーム スイッチのみ） 88
IPv6 ネイバー探索の構成 89
オプションの IPv6 ネイバー探索 92
IPv6 パケット検証の構成 93
IPv6 ステートレス自動構成の定義 95
LPM ヘビー ルーティング モードの設定（Cisco Nexus 9200 および 9300-EX プラットフォーム スイッチおよび 9732C-EX ライン カードのみ） 96
LPM インターネット ピアリング ルーティング モードの設定（Cisco Nexus 9500-R プラットフォーム スイッチ、Cisco Nexus 9300-EX プラットフォーム スイッチ、および Cisco Nexus 9000 シリーズ スイッチと 9700-EX ライン カードのみ） 97
LPM インターネット ピアリング ルーティング モードの追加設定 99
LPM デュアルホスト ルーティング モードの設定（Cisco Nexus 9200 および 9300-EX プラットフォーム スイッチ） 101
IPv6 リダイレクト Syslog の構成 102
IPv6 設定の確認 103
IPv6 の設定例 103

CHAPTER 5
DNS の設定 105
DNS クライアントについて 105
DNS クライアントの概要 105
ネーム サーバ 106
DNS の動作 106
高可用性 106
仮想化のサポート 106
DNS クライアントの前提条件 107
DNS クライアントに関する注意事項と制約事項 107
DNS クライアントのデフォルト設定 107
DNS クライアントの設定 107
DNS クライアントの設定 107

仮想化の設定 110
DNS クライアントの設定の確認 112
DNS クライアントの設定例 112

CHAPTER 6

OSPFv2 の設定 113
OSPFv2 について 113
OSPFv2およびユニキャストRIB 114
認証 114
簡易パスワード認証 115
暗号化認証 115
MD5 認証 115
HMAC-SHA 認証 115
高度な機能 116
スタブ エリア 116
Not So Stubby Area 116
仮想リンク 117
ルートの再配布 118
ルート集約 118
高可用性およびグレースフル リスタート 119
OSPFv2 スタブ ルータ アドバタイズメント 119
複数の OSPFv2 インスタンス 120
SPF 最適化 120
BFD 120
OSPFv2 の仮想化のサポート 120
OSPFv2 の前提条件 121
OSPFv2 の注意事項および制約事項 121
OSPFv2のデフォルト設定 123
基本的な OSPFv2 の設定 124
OSPFv2の有効化 124
OSPFv2インスタンスの作成 125
OSPFv2 インスタンスのオプション パラメータの設定 126

OSPFv2でのネットワークの設定 128
エリアの認証の設定 131
インターフェイスの認証の設定 133
高度なOSPFv2の設定 136
境界ルータのフィルタ リストの設定 136
スタブ エリアの設定 138
Totally Stubby エリアの設定 139
NSSA の設定 140
マルチエリアの隣接関係の設定 143
仮想リンクの設定 144
再配布の設定 147
再配布されるルート数の制限 149
ルート集約の設定 151
スタブ ルート アドバタイズメントの設定 153
ルートのアドミニストレーティブ ディスタンスの設定 154
デフォルト タイマーの変更 157
グレースフル リスタートの設定 160
OSPFv2 インスタンスの再起動 162
仮想化による OSPFv2 の設定 163
OSPFv2 設定の確認 165
OSPFv2 のモニタリング 166
OSPFv2 の設定例 166
OSPF RFC 互換モードの例 167
その他の参考資料 167
OSPFv2 の関連資料 167
MIB 167

CHAPTER 7
OSPFv3 の設定 169
OSPFv3 について 169
OSPFv3 と OSPFv2 の比較 170
Hello パケット 170

ネイバー情報 171
隣接関係 172
指定ルータ 172
エリア 173
リンクステート アドバタイズメント 174
リンクステート アドバタイズメント タイプ 174
リンク コスト 175
フラッディングと LSA グループ ペーシング 176
リンクステート データベース 176
マルチエリア隣接関係（Multi-Area Adjacency） 177
OSPFv3 と IPv6 ユニキャスト RIB 177
アドレス ファミリのサポート 177
認証および暗号化 178
高度な機能 178
スタブ エリア 178
Not-So-Stubby Area 179
仮想リンク 180
ルートの再配布 180
ルート集約 181
高可用性およびグレースフル リスタート 181
複数の OSPFv3 インスタンス 182
SPF 最適化 182
BFD 183
仮想化のサポート 183
OSPFv3 の前提条件 183
OSPFv3 の注意事項および制約事項 184
デフォルト設定 186
基本的なOSPFv3の設定 186
OSPFv3の有効化 186
OSPFv3インスタンスの作成 187
OSPFv3でのネットワークの設定 190

高度なOSPFv3の設定 193
境界ルータのフィルタ リストの設定 193
スタブ エリアの設定 195
Totally Stubby エリアの設定 196
NSSA の設定 197
マルチエリアの隣接関係の設定 199
仮想リンクの設定 201
再配布の設定 203
再配布されるルート数の制限 205
ルート集約の設定 207
ルートのアドミニストレーティブ ディスタンスの設定 209
デフォルト タイマーの変更 212
グレースフル リスタートの設定 215
OSPFv3 インスタンスの再起動 216
仮想化による OSPFv3 の設定 217
暗号化および認証の構成 219
ルータ レベルでの OSPFv3 暗号化の設定 220
エリア レベルでの OSPFv3 暗号化の設定 221
インターフェイスレベルでの OSPFv3 暗号化の設定 222
仮想リンクの OSPFv3 暗号化の設定 224
ルータ レベルで OSPFv3 認証の構成 226
エリア レベルで OSPFv3 認証の構成 227
インターフェイス レベルで OSPFv3 認証の構成 229
仮想リンク レベルでの OSPFv3 認証の構成 231
OSPFv3 の設定の確認 232
OSPFv3のモニタリング 234
OSPFv3 の設定例 234
関連項目 235
その他の参考資料 235
MIB 235

CHAPTER 8 EIGRP の設定 237
EIGRP について 237
EIGRP コンポーネント 237
信頼性の高いトランスポート プロトコル 238
ネイバー探索およびネイバー回復 238
拡散更新アルゴリズム 239
EIGRP ルート更新 239
内部ルート メトリック 239
ワイド メトリックス 240
外部ルート メトリック 241
EIGRP とユニキャスト RIB 241
高度な EIGRP 241
アドレス ファミリ 241
認証 242
スタブ ルータ 242
ルート集約 243
ルートの再配布 243
ロード バランシング 243
Split Horizon 244
BFD 244
仮想化のサポート 244
グレースフル リスタートおよびハイ アベイラビリティ 245
複数の EIGRP インスタンス 246
EIGRP の前提条件 246
EIGRP の注意事項と制約事項 246
デフォルト設定 248
基本的な EIGRP の設定 249
EIGRP 機能の有効化 249
EIGRP インスタンスの作成 249
EIGRP インスタンスの再起動 252

EIGRP インスタンスのシャットダウン 253
EIGRP のパッシブ インターフェイスの設定 253
インターフェイスでの EIGRP のシャットダウン 254
高度な EIGRP の設定 255
EIGRP での認証の設定 255
EIGRP スタブ ルーティングの設定 257
EIGRP のサマリー アドレスの設定 258
EIGRP へのルートの再配布 259
再配布されるルート数の制限 261
EIGRP でのロードバランスの設定 264
EIGRP のグレースフル リスタートの設定 265
hello パケット間のインターバルとホールド タイムの調整 267
スプリット ホライズンの無効化 268
ワイド メトリックスの有効化 269
EIGRP の調整 269
EIGRP の仮想化の設定 273
EIGRP の設定の確認 274
EIGRP のモニタリング 275
EIGRP の設定例 275
関連項目 276
その他の参考資料 276
関連資料 277
MIB 277

CHAPTER 9

IS-IS の設定 279
IS-IS について 279
IS-IS の概要 280
IS-IS エリア 280
NET およびシステム ID 281
DIS 281
IS-IS 認証 282

メッシュ グループ 282
過負荷ビット 283
ルート集約 283
ルートの再配布 283
プレフィックスの抑制のリンク 284
ロード バランシング 284
BFD 284
仮想化のサポート 284
高可用性およびグレースフル リスタート 285
複数の IS-IS インスタンス 285
IS-IS の前提条件 285
IS-IS に関する注意事項および制限事項 286
デフォルト設定 286
IS-IS の設定 287
IS-IS コンフィギュレーション モード 287
IS-IS 機能の有効化 288
IS-IS インスタンスの作成 288
IS-IS インスタンスの再起動 291
IS-IS のシャットダウン 291
インターフェイスでの IS-IS の設定 292
インターフェイスでの IS-IS のシャットダウン 293
エリアでの IS-IS 認証の設定 294
インターフェイスでの IS-IS 認証の設定 295
メッシュ グループの設定 297
指定中継システムの設定 297
ダイナミック ホスト交換の設定 298
過負荷ビットの設定 298
接続ビットの設定 299
hello パディングの一時モードの設定 299
サマリー アドレスの設定 300
再配布の設定 301

再配布されるルート数の制限 303
パッシブインターフェイスプレフィックスのみのアドバタイズ 305
インターフェイスでのプレフィックスの抑制 306
厳密な隣接モードのディセーブル化 307
グレースフル リスタートの設定 309
仮想化の設定 310
IS-IS の調整 313
IS-IS 設定の確認 315
IS-IS の監視 316
IS-IS の設定例 317
関連項目 317

CHAPTER 10 基本的 BGP の設定 319
基本的な BGP について 319
BGP 自律システム 320
4 バイトの AS 番号のサポート 320
アドミニストレーティブ ディスタンス 320
BGP ピア 321
BGP セッション 321
プレフィックス ピアおよびインターフェイス ピアのダイナミック AS 番号 321
BGP ルータ ID 322
BGP パスの選択 322
BGP パス選択：パスびペアの比較 323
BGPパス選択：比較の順序の決定 325
BGP パス選択：最適パス変更抑制の決定 326
BGP およびユニキャスト RIB 326
BGP プレフィックス独立コンバージェンス 327
BGP PIC エッジ ユニパス 327
マルチパスを持つ BGP PIC エッジ 328
BGP PIC コア 329
BGP PIC の機能サポート マトリクス 329

BGP の仮想化 330
BGP の前提条件 330
基本 BGP に関する注意事項と制約事項 330
デフォルト設定 332
CLI コンフィギュレーション モード 333
グローバル コンフィギュレーション モード 333
アドレス ファミリ設定モード 333
ネイバー コンフィギュレーション モード 334
ネイバー アドレス ファミリ コンフィギュレーション モード 334
基本的 BGP の設定 335
BGPの有効化 335
BGP インスタンスの作成 336
BGP インスタンスの再起動 338
BGP のシャットダウン 338
BGP ピア設定 339
プレフィックス ピアのダイナミック AS 番号の設定 341
BGP PIC エッジの設定 343
BGP PIC コアの設定 345
BGP 情報の消去 346
ベーシック BGP の設定の確認 351
BGP 統計情報のモニタリング 354
ベーシック BGP の設定例 354
関連項目 354
次の作業 354
その他の参考資料 355
ベーシック BGP の MIB 355

CHAPTER 11 高度な **BGP** の設定 357
拡張 BGP について 358
ピア テンプレート 358
認証 359

ルート ポリシーおよび BGP セッションのリセット 359
eBGP 360
iBGP 360
AS 連合 361
ルート リフレクタ 361
機能ネゴシエーション 362
ルート ダンプニング 362
ロード シェアリングおよびマルチパス 363
BGP の追加パス 364
ルート集約 364
BGP 条件付きアドバタイズメント 365
BGP ネクスト ホップ アドレス トラッキング 365
ルートの再配布 366
ラベル付きユニキャスト ルートとラベルなしユニキャスト ルート 366
BFD 367
BGP の調整 367
BGP タイマー 367
ベストパス アルゴリズムの調整 367
マルチプロトコル BGP 368
RFC 5549 368
RFC 6368 368
BGP モニタリング プロトコル 370
グレースフル リスタートおよびハイ アベイラビリティ 370
メモリ不足の処理 371
仮想化のサポート 372
拡張 BGP の前提条件 372
拡張 BGP に関する注意事項と制限事項 372
デフォルト設定 378
高度な BGP の設定 378
インターフェイスでの IP 転送の有効化 378
BGP セッション テンプレートの設定 379
BGP peer-policy テンプレートの設定 382

BGP peer テンプレートの設定 384
プレフィックス ピアリングの設定 387
IPv4 および IPv6 アドレス ファミリ向け IPv6 リンク ローカル経由の BGP インターフェイス ピアリングの設定 389
BGP 認証の設定 392
BGP セッションのリセット 394
ネクストホップ アドレスの変更 395
BGP ネクスト ホップ アドレス トラッキングの設定 396
ネクスト ホップ フィルタリングの設定 397
デフォルト ルートによるネクストホップ解決の設定 397
ネクストホップセルフによるリフレクト ルートの制御 398
セッションがダウンした場合のネクストホップ グループの縮小 398
機能ネゴシエーションのディセーブル化 399
ポリシーのバッチ処理の無効化 400
BGP 追加パスの設定 400
追加パスの送受信機能のアドバタイズ 400
追加パスの送受信の設定 401
アドバタイズされるパスの設定 402
追加パス選択の設定 404
eBGP の設定 405
eBGP シングルホップ チェックの無効化 405
TTL セキュリティ ホップの構成 405
eBGP マルチホップの設定 408
高速外部フォールオーバーの無効化 409
AS パス属性の制限 409
ローカル AS サポートの設定 410
AS 連合の設定 410
ルート リフレクタの設定 411
アウトバウンド ルート マップを使用した、反映されたルートのネクスト ホップの設定 414
ルート ダンプニングの設定 417
ロード シェアリングおよび ECMP の設定 417

BGP 経由不等コストマルチパス（UCMP） 418
UCMP over BGP の有効化 418
BGP 経由 UCMP の注意事項と制限事項 419
最大プレフィックス数の設定 419
DSCP の設定 420
ダイナミック機能の設定 420
集約アドレスの設定 421
BGP ルートの抑制 422
BGP 条件付きアドバタイズメントの設定 423
ルートの再配布の設定 426
DMZ リンク帯域幅 427
注意事項と制約事項 428
BGP DMZ リンク帯域幅の構成 428
BGP DMZ リンク帯域幅の構成例 430
リンク帯域幅拡張コミュニティを使用した不等コスト マルチパス（UCMP）の構成 432
設定例 434
設定の確認 438
デフォルト ルートのアドバタイズ 439
BGP 属性フィルタリングの設定とエラー処理 440
BGP 更新メッセージからのパス属性の取り消しとしての処理 441
BGP 更新メッセージからのパス属性の破棄 441
拡張属性エラー処理のイネーブル化またはディセーブル化 442
取り消されたパス属性または破棄されたパス属性の表示 442
BGP の調整 443
ポリシーベースのアドミニストレーティブ ディスタンスの設定 449
マルチプロトコル BGP の設定 451
BMP の設定 452
BGP ローカル ルート リーク 455
BGP ローカル ルート リークについて 455
BGP ローカル ルート リークの注意事項と制約事項 455
デフォルト VRF にリークするために VPN からインポートされたルートを設定する 456

デフォルト VRF からリークされたルートを VPN にエクスポートするための設定 457
VRF にエクスポートするために VPN からインポートしたルートの設定 458
VRF からインポートして VPN にエクスポートするルートの設定 459
設定例 460
BGP ローカル ルート リーク情報の表示 463
BGP グレースフル シャットダウン 464
BGP グレース フル シャットダウンに関する情報 464
グレースフル シャットダウンの認識とアクティブ化 464
グレースフル シャットダウンのコンテキスト 465
ルート マップによるグレースフル シャットダウン 465
注意事項と制約事項 467
グレースフル シャットダウン タスクの概要 468
リンクのグレースフル シャットダウンの設定 468
GRACEFUL_SHUTDOWN コミュニティに基づく BGP ルートのフィルタリングとローカル プリファレンスの設定 469
すべての BGP ネイバーのグレースフル シャットダウンの設定 471
GRACEFUL_SHUTDOWN コミュニティを使用したすべてのルートのプリファレンスの制御 473
GRACEFUL_SHUTDOWN コミュニティのピアへの送信の防止 474
グレースフル シャットダウン情報の表示 474
グレースフル シャットダウンの設定例 475
グレースフル リスタートの設定 477
仮想化の設定 480
拡張 BGP の設定の確認 482
BGP 統計情報のモニタリング 484
設定例 485
関連項目 486
その他の参考資料 486
MIB 486

CHAPTER 12 **RIP の設定** 487

RIP について 487
RIP の概要 487
RIPv2 認証 488
Split Horizon 488
ルートのフィルタリング 489
ルート集約 489
ルートの再配布 489
ロード バランシング 490
RIP のハイ アベイラビリティ 490
RIP 仮想化のサポート 490
RIP の前提条件 490
RIP に関する注意事項と制約事項 490
RIP パラメータのデフォルト設定 491
RIP の設定 491
RIP のイネーブル化 491
RIP インスタンスの作成 492
RIP インスタンスの再起動 494
インターフェイスでの RIP の設定 494
RIP 認証の設定 495
パッシブ インターフェイスの設定 497
ポイズン リバースを指定したスプリット ホライズンの設定 497
ルート集約の設定 498
ルートの再配布の設定 498
Cisco IOS RIP との互換性のため、Cisco NX-OS RIP を設定 500
仮想化の設定 503
RIP の調整 505
RIP の設定の確認 507
RIP 統計情報の表示 507
RIP の設定例 507
関連項目 508

CHAPTER 13 RIPng の設定 509

RIPng について 509

RIPng の概要 509

Split Horizon 510

ルートのフィルタリング 510

ロード バランシング 510

デフォルトの情報の発信元と生成 511

RIPng の高可用性 511

RIPng の仮想化のサポート 511

RIPng の前提条件 511

RIPng のガイドラインと制限事項 512

RIPng パラメータのデフォルト設定 512

RIPng の設定 512

RIPng の有効化 512

RIPng インスタンスの作成 513

RIPng インスタンスの再起動 515

インターフェイス上での RIPng の構成 515

ポイズン リバースを指定したスプリット ホライズンの設定 516

Cisco IOS RIPng との互換性のための Cisco NX-OS RIPng の構成 517

仮想化の設定 519

RIPng のチューニング 521

RIPng 構成の確認 522

RIPng 統計の表示 523

RIPng の設定例 523

関連項目 523

CHAPTER 14 スタティック ルーティングの設定 525

スタティック ルーティングについて 525

アドミニストレーティブ ディスタンス 526

直接接続のスタティック ルート 526

完全指定のスタティック ルート 526
フローティング スタティック ルート 526
スタティック ルートのリモート ネクスト ホップ 527
BFD 527
仮想化のサポート 527
スタティック ルーティングの前提条件 527
デフォルト設定 527
スタティック ルーティングの設定 528
スタティック ルーティングの設定 528
VLAN を介したスタティック ルートの設定 529
仮想化の設定 531
スタティック ルーティングの設定確認 532
スタティック ルーティングの設定例 533

CHAPTER 15 レイヤ 3 仮想化の設定 535
レイヤ 3 仮想化について 535
VRF およびルーティング 536
デフォルトの VRF からのルート リークとルートのインポート 537
IPv6 専用環境の BGP VRF ルーター ID 537
VRF 認識サービス 538
Reachability 539
フィルタリング 539
到達可能性とフィルタリングの組み合わせ 540
VRF の前提条件 540
VRF の注意事項および制約事項 540
VRF ルート リークの注意事項と制約事項 541
デフォルト設定 542
VRF の設定 542
VRF の作成 542
インターフェイスへの VRF メンバーシップの割当て 544
ルーティング プロトコル用の VRF パラメータの設定 545

VRF 認識サービスの設定 547
VRF スコープの設定 549
VRF の設定の確認 550
VRF の設定例 550
その他の参考資料 557
VRF の関連資料 557
標準 557

CHAPTER 16
ユニキャスト RIB および FIB の管理 559
ユニキャスト RIB および FIB について 559
レイヤ 3 整合性チェッカー 560
ユニキャスト RIB に関する注意事項と制約事項 560
ユニキャスト RIB および FIB の管理 561
モジュールの FIB 情報の表示 561
ユニキャスト FIB でのロード シェアリングの設定 562
ダイナミック ロード バランシング 565
ダイナミックロードバランシングについて 565
ダイナミック ロード バランシングの主要な概念 567
ダイナミックロードバランシングに関するガイドラインと制限事項 569
ダイナミック ロード バランシングの構成 571
ダイナミックロードバランシングの構成例 576
ダイナミックロードバランシングの確認 576
ダイナミックロードバランシングのトラブルシュート 578
ルーティング情報と隣接情報の表示 579
レイヤ 3 整合性チェッカーのトリガー 580
FIB 内の転送情報の消去 581
ユニキャスト RIB の最大ルート数の設定 582
ルートのメモリ要件の見積もり 583
ユニキャスト RIB 内のルートの消去 583
ユニキャスト RIB および FIB の確認 584
その他の参考資料 585

関連資料 585

CHAPTER 17

Route Policy Manager の設定 587

Route Policy Manager について 587

プレフィックス リスト 587

MAC リスト 588

ルート マップ 588

ルートマップのシーケンスのデフォルトアクション 589

ルート マップのデフォルト シーケンス番号 589

一致基準 589

設定変更 590

アクセス リスト 590

BGP の AS 番号 590

BGP の AS パス リスト 590

BGP のコミュニティ リスト 591

BGP の拡張コミュニティ リスト 591

NX-OS BGP の大規模コミュニティの構成 592

ルートの再配布およびルート マップ 597

Route Policy Manager の注意事項と制約事項 598

Route Policy Manager パラメータのデフォルト設定 599

Route Policy Manager の設定 600

IP プレフィックス リストの設定 600

MAC リストの設定 602

AS パス リストの設定 604

BGP AS-path 属性の置き換え 605

完全な AS パスの置き換え 606

AS パスでの選択した AS 番号の置き換え 607

コミュニティ リストの設定 608

拡張コミュニティ リストの設定 610

ルート マップの設定 612

ルート マップの削除をブロックするグローバル コマンド 621

Route Policy Manager の設定の確認 621
Route Policy Manager の設定例 622
関連項目 622

CHAPTER 18 ポリシーベース ルーティングの設定 623
ポリシーベース ルーティングについて 623
ポリシー ルート マップ 624
ポリシーベース ルーティングの set 基準 624
ポリシーベースルーティングのルートマップサポートマトリックス 625
ルート マップ処理ロジック 626
ポリシーベース ルーティングの前提条件 627
ポリシーベース ルーティングの注意事項と制約事項 627
ポリシーベース ルーティングのデフォルト設定 630
ポリシーベース ルーティングの設定 631
ポリシーベース ルーティング機能のイネーブル化 631
ECMP 上のポリシーベース ルーティングの有効化 632
PBR 高速コンバージェンスの設定 633
ルート ポリシーの設定 635
ルートマップを使用した VRF へのリダイレクト 640
VRF の構成 641
ネクストホップに一致するデフォルト ルートをリダイレクト 643
ポリシーベース ルーティングの設定の確認 645
ポリシーベース ルーティングの設定例 646
ポリシーベースルーティングの関連資料 649

CHAPTER 19 『Configuring HSRP』 651
HSRP について 651
HSRP の概要 652
HSRP のバージョン 653
HSRP for IPv4 654
HSRP for IPv6。 654

IPv6 アドレスの HSRP 655
HSRP サブネット VIP 656
HSRP 認証 656
HSRP メッセージ 656
HSRP ロード シェアリング 657
オブジェクト トラッキングおよび HSRP 658
vPC と HSRP 658
vPC ピア ゲートウェイと HSRP 658
BFD 658
ハイ アベイラビリティおよび拡張ノンストップ フォワーディング 659
仮想化のサポート 659
HSRP の前提条件 659
HSRP の注意事項と制約事項 659
HSRP パラメータのデフォルト設定 662
『Configuring HSRP』 662
HSRP の有効化 662
HSRP バージョン設定 663
IPv4 の HSRP グループの設定 663
IPv6 の HSRP グループの設定 665
HSRP 仮想 MAC アドレスの設定 667
HSRP の認証 668
HSRP オブジェクト トラッキングの設定 670
HSRP プライオリティの設定 673
HSRP コンフィギュレーション モードでの HSRP のカスタマイズ 674
インターフェイスコンフィギュレーションモードでのHSRPのカスタマイズ 676
HSRP の拡張ホールド タイマーの設定 677
HSRP 設定の確認 678
HSRP の設定例 679
その他の参考資料 680
関連資料 680
MIB 680

CHAPTER 20

VRRP の設定 681

VRRP について 681

VRRP の動作 682

VRRP の利点 683

複数の VRRP グループ 684

VRRP ルータのプライオリティおよびプリエンプション 685

vPC と VRRP 685

VRRP のアドバタイズメント 686

VRRP 認証 686

VRRP トラッキング 686

VRRP 用 BFD 687

VRRPv3およびVRRSに関する情報 687

VRRPv3 の利点 688

VRRPv3 オブジェクト トラッキング 688

高可用性 688

仮想化のサポート 688

VRRP の注意事項と制約事項 689

VRRPv3 の注意事項および制約事項 689

VRRP パラメータのデフォルト設定 690

VRRPv3 パラメータのデフォルト設定 691

VRRP の設定 691

VRRP のイネーブル化 691

VRRP グループの設定 692

VRRP プライオリティの設定 694

VRRP 認証の設定 695

アドバタイズメント パケットのタイム インターバルの設定 697

プリエンプションのディセーブル化 698

VRRP インターフェイス ステート トラッキングの設定 700

VRRP オブジェクト トラッキングの設定 702

VRRPv3 の設定 703

VRRPv3 および VRRS の有効化 703
VRRPv3 グループの作成 704
VRRPv3 コントロールグループの設定 706
VRRPv3 オブジェクト トラッキングの設定 708
VRRS 経路の設定 710
VRRP の設定の確認 711
VRRPv3 設定の確認 711
VRRP 統計情報のモニタリングとクリア 712
VRRPv3 統計情報のモニタリングとクリア 712
VRRP の設定例 712
VRRPv3 の設定例 714
その他の参考資料 715
VRRP の関連資料 715

CHAPTER 21 オブジェクト トラッキングの設定 717
オブジェクト トラッキングについて 717
オブジェクト トラッキングの概要 717
オブジェクト トラッキング リスト 718
高可用性 719
仮想化のサポート 719
オブジェクト トラッキングの設定例 719
オブジェクト トラッキングに関する注意事項と制約事項 719
デフォルト設定 720
オブジェクト トラッキングの設定 720
インターフェイスに対するオブジェクト トラッキングの設定 720
トラッキング オブジェクトの削除 721
ルート到達可能性に対するオブジェクト トラッキングの設定 722
ブール式を含むオブジェクト トラッキング リストの設定 723
パーセンテージしきい値を含むオブジェクト トラッキング リストの設定 725
重みしきい値を含むオブジェクト トラッキング リストの設定 727
オブジェクト トラッキングの遅延の設定 728

非デフォルト VRF に対するオブジェクト トラッキングの設定 731
オブジェクト トラッキングの設定の確認 732
オブジェクト トラッキングの設定例 732
関連項目 733
その他の参考資料 733
関連資料 733

APPENDIX A
Cisco NX-OS ユニキャスト機能でサポートされている IETF RFC 735
BGP の RFC 735
ファーストホップ冗長プロトコルの RFC 737
IP サービスに関する RFC の参考資料 737
IPv6 の RFC 737
IS-IS の RFC 738
OSPF の RFC 739
RIP の RFC 739

APPENDIX B
Cisco NX-OS レイヤ 3 ユニキャスト機能の設定の制限 741
Cisco NX-OS レイヤ 3 ユニキャスト機能の設定の制限 741

はじめに

この前書きは、次の項で構成されています。

- 対象読者 （xxxiii ページ）
- 表記法 （xxxiii ページ）
- Cisco Nexus 9000 シリーズ スイッチの関連資料 （xxxiv ページ）
- マニュアルに関するフィードバック （xxxiv ページ）
- 通信、サービス、およびその他の情報 （xxxv ページ）

対象読者

このマニュアルは、Cisco Nexus スイッチの設置、設定、および維持に携わるネットワーク管理者を対象としています。

表記法

コマンドの説明には、次のような表記法が使用されます。

表記法	説明
bold	太字の文字は、表示どおりにユーザが入力するコマンドおよびキーワードです。
italic	イタリック体の文字は、ユーザが値を指定する引数です。
[x]	省略可能な要素（キーワードまたは引数）は、角かっこで囲んで示しています。
[x | y]	いずれか 1 つを選択できる省略可能なキーワードや引数は、角かっこで囲み、縦棒で区切って示しています。
{x | y}	必ずいずれか1つを選択しなければならない必須キーワードや引数は、波かっこで囲み、縦棒で区切って示しています。

表記法	説明
[x {y | z}]	角かっこまたは波かっこが入れ子になっている箇所は、任意または必須の要素内の任意または必須の選択肢であることを表します。角かっこ内の波かっこと縦棒は、省略可能な要素内で選択すべき必須の要素を示しています。
`variable`	ユーザが値を入力する変数であることを表します。イタリック体が使用できない場合に使用されます。
string	引用符を付けない一組の文字。string の前後には引用符を使用しないでください。引用符を使用すると、その引用符も含めて string と見なされます。

例では、次の表記法を使用しています。

表記法	説明
`screen` フォント	スイッチが表示する端末セッションおよび情報は、スクリーンフォントで示しています。
太字の `screen` フォント	ユーザが入力しなければならない情報は、太字の screen フォントで示しています。
イタリック体の *screen* フォント	ユーザが値を指定する引数は、イタリック体の screen フォントで示しています。
<>	パスワードのように出力されない文字は、山カッコ（<>）で囲んで示しています。
[ ]	システム プロンプトに対するデフォルトの応答は、角カッコで囲んで示しています。
!、#	コードの先頭に感嘆符（!）またはポンド記号（#）がある場合には、コメント行であることを示します。

Cisco Nexus 9000 シリーズ スイッチの関連資料

Cisco Nexus 9000 シリーズ スイッチ全体のマニュアル セットは、次の URL にあります。

https://www.cisco.com/en/US/products/ps13386/tsd_products_support_series_home.html

マニュアルに関するフィードバック

このマニュアルに関する技術的なフィードバック、または誤りや記載もれなどお気づきの点がございましたら、HTML ドキュメント内のフィードバック フォームよりご連絡ください。ご協力をよろしくお願いいたします。

通信、サービス、およびその他の情報

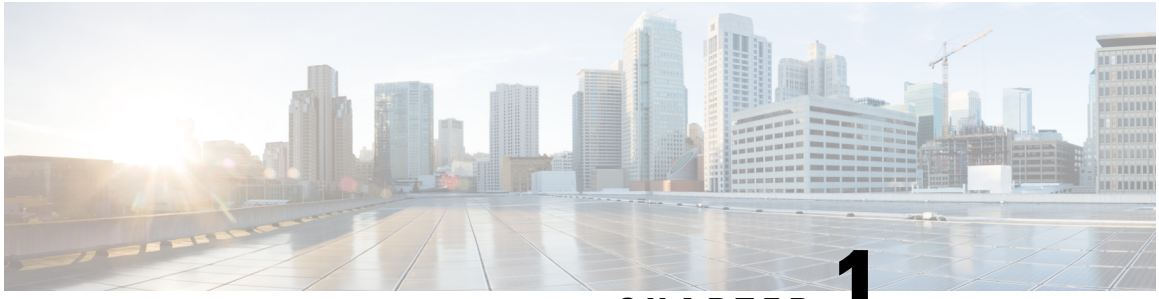
- シスコからタイムリーな関連情報を受け取るには、Cisco Profile Manager でサインアップしてください。
- 重要な技術によりビジネスに必要な影響を与えるには、Cisco Services にアクセスしてください。
- サービス リクエストを送信するには、Cisco Support にアクセスしてください。
- 安全で検証済みのエンタープライズクラスのアプリケーション、製品、ソリューション、およびサービスを探して参照するには、Cisco DevNet [英語] にアクセスしてください。
- 一般的なネットワーキング、トレーニング、認定関連の出版物を入手するには、Cisco Press にアクセスしてください。
- 特定の製品または製品ファミリの保証情報を探すには、Cisco Warranty Finder にアクセスしてください。

Cisco バグ検索ツール

シスコバグ検索ツール（BST）は、シスコ製品とソフトウェアの障害と脆弱性の包括的なリストを管理するシスコバグ追跡システムへのゲートウェイです。BST は、製品とソフトウェアに関する詳細な障害情報を提供します。

マニュアルに関するフィードバック

シスコのテクニカルドキュメントに関するフィードバックを提供するには、それぞれのオンラインドキュメントの右側のペインにあるフィードバックフォームを使用してください。



CHAPTER 1

New and Changed Information

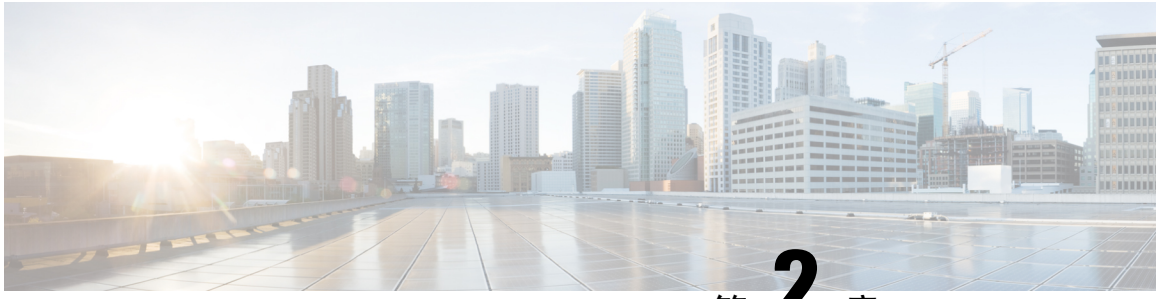
- New and Changed Information, on page 1

New and Changed Information

Table 1: New and Changed Features

Feature	Description	Changed in Release	Where Documented
Next-hop resolution using VRF	The **set vrf** command in a route-map allows you to specify the VRF context that should be used for route lookups or next-hop resolution. This command allows you to control which VRF is used to lookup routes, rather than defaulting to the VRF associated with the ingress interface.	10.5(2)F	ポリシーベース ルーティングの set 基準, on page 624 ポリシーベースルーティングのルートマップサポートマトリックス, on page 625 ポリシーベース ルーティングの注意事項と制約事項, on page 627 ルート ポリシーの設定, on page 635 ルートマップを使用した VRF へのリダイレクト, on page 640
ECMP Dynamic Load Balancing (DLB)	Provides for Layer 3 ECMP dynamic load balancing based on current load on the link	10.5(1)F	ダイナミック ロードバランシング, on page 565

Feature	Description	Changed in Release	Where Documented
ECMP Dynamic Load Balancing Static Pinning	Support static-pinning where a source port can be pinned to a destination port which is part of a DLB Enabled ECMP Group. All the traffic from this source port will be sent to the pinned destination port.	10.5(1)F	ダイナミック ロードバランシング, on page 565
Exclude-L3-Proto in ECMP hashing	Added the **exclude-l3-proto** option to the **ip load-sharing address** command to allow the exclusion of the IP protocol from ECMP hashing during next-hop selection on Cisco Nexus 9300-GX2 Series switches.	10.5(1)F	ユニキャスト FIB でのロード シェアリングの設定, on page 562
Detailed information added in show commands for routes advertised to or received from a specific neighbor.	Added the **detail** to the **show bgp {ipv4 | ipv6} neighbors {<ip-address> | <ipv6-prefix>} routes {advertised | received} [vrf <vrf-name>]** detailed information of all routes advertised to or received from a specific neighbor.	10.5(1)F	基本的 BGP の設定, on page 319
Unequal Cost Multi Path Load balancing with BGP Link bandwidth Extended Community attribute	Added new features to support Unequal Cost Multi Path Load balancing with BGP Link bandwidth Extended Community attribute.	10.5(1)F	DMZ リンク帯域幅, on page 427



第 2 章

概要

この章は、次の項で構成されています。

- ライセンス要件 （3 ページ）
- サポートされるプラットフォーム （3 ページ）
- レイヤ 3 ユニキャスト ルーティングについて （3 ページ）
- ルーティング アルゴリズム （10 ページ）
- レイヤ 3 仮想化 （12 ページ）
- Cisco NX-OS フォワーディング アーキテクチャ （12 ページ）
- レイヤ 3 ユニキャスト ルーティング機能のまとめ （15 ページ）
- 関連項目 （18 ページ）

ライセンス要件

Cisco NX-OS ライセンス方式の推奨の詳細と、ライセンスの取得および適用の方法については、『*Cisco NX-OS ライセンス ガイド*』および『*Cisco NX-OS ライセンス オプション ガイド*』を参照してください。

サポートされるプラットフォーム

Cisco NX-OS リリース 7.0(3)I7(1) 以降では、Nexus スイッチ プラットフォーム サポート マトリクスに基づいて、選択した機能をさまざまな Cisco Nexus 9000 および 3000 スイッチで使用するために、どの Cisco NX-OS リリースが必要かを確認してください。

レイヤ 3 ユニキャスト ルーティングについて

レイヤ 3 ユニキャスト ルーティングには 2 つの基本的動作（最適なルーティング パスの決定およびパケットの交換）があります。ルーティング アルゴリズムを使用すると、ルータから宛先までの最適なパス（経路）を計算できます。この計算方法は、選択したアルゴリズム、ルート メトリック、そしてロード バランシングや代替パスの探索などの考慮事項により異なります。

ルーティングの基礎

ルーティング プロトコルは、メトリックを使用して、宛先までの最適なパスを調べます。メトリックとは、パス帯域幅などの、ルーティング アルゴリズムが宛先までの最適なパスを決定するために使用する測定基準です。パスを決定しやすいように、ルーティング アルゴリズムは、ルート情報（IP 宛先アドレス、次のルータまたはネクスト ホップのアドレスなど）を含むルーティング テーブルを初期化して維持します。宛先とネクスト ホップの関連付けにより、ルータは、宛先までの途中にあるネクスト ホップとなる特定のルータにパケットを送信すると、最適なパスで IP 宛先まで届けられることを判定できます。ルータは、着信パケットを受信すると、宛先アドレスをチェックし、このアドレスをネクスト ホップと関連付けようとします。ルート テーブルの詳細については、「ユニキャスト RIB」の項を参照してください。

ルーティング テーブルには、パスの優先度に関するデータなど、その他の情報が含まれていることもあります。ルータは、メトリックを比較して最適なルートを決定します。これらのメトリックは、使用しているルーティング アルゴリズムの設計によって異なります。「ルーティング メトリック」の項を参照してください。

各ルータは互いに通信し、さまざまなメッセージを送信して、そのルーティング テーブルを維持します。ルーティング更新メッセージは、ルーティング テーブルの全部または一部で構成されるメッセージです。ルータは、他のすべてのルータからのルーティング更新情報を分析して、ネットワーク トポロジの詳細な図を構築できます。ルータ間で送信されるメッセージのもう 1 つの例であるリンク ステート アドバタイズメントは、送信ルータのリンク状態を他のルータに通知します。リンク情報を使用して、ルータが、ネットワーク宛先までの最適なルートを決定できるようにすることもできます。詳細については、「ルーティング アルゴリズム」の項を参照してください。

パケット交換

パケット交換では、ホストが、パケットを別のホストに送信する必要があることを決定します。何らかの手段でルータ アドレスを取得したら、送信元ホストは、明確にルータの物理（メディア アクセス コントロール（MAC）レイヤ）アドレスにアドレス指定されているが、宛先ホストの IP（ネットワーク層）アドレスを含むパケットを送信します。

ルータは宛先の IP アドレスを調べ、ルーティング テーブルでその IP アドレスを探します。ルータがパケットの転送方法を認識していない場合は、通常はパケットをドロップします。パケットの転送方法がわかった場合、ルータは、宛先の MAC アドレスをネクスト ホップ ルータの MAC アドレスに変更し、パケットを送信します。

ネクスト ホップが宛先のホストである場合や、同じ交換決定処理を行う別のルータである場合があります。パケットがインターネットワークを介して移動するにつれ、パケットの物理アドレスは変化しますが、プロトコル アドレスは一定のままです（次の図を参照）。

図 *1*: ネットワークを介したパケット ヘッダーの更新

ルーティング メトリック

ルーティング アルゴリズムは、多くの異なるメトリックを使用して最適なルートを決定します。高度なルーティング アルゴリズムは、複数のメトリックに基づいてルートを選択している場合があります。

パス長

パスの長さは、最も一般的なルーティング メトリックです。一部のルーティング プロトコルでは、各ネットワーク リンクに恣意的なコストの割り当てが可能です。この場合、パスの長さは、経由した各リンクに関連付けられたコストの合計となります。それ以外のルーティング プロトコルでは、パケットが送信元から宛先までに経由する必要のある、ルータなどのネットワーク間製品の通過回数を指定するメトリックであるホップ数が定義されます。

Reliability

ルーティングアルゴリズムとの関連における信頼性は、各ネットワーク リンクの信頼性（ビット誤り率で示される）です。一部のネットワーク リンクは、他のネットワーク リンクよりダウンする頻度が高い場合があります。ネットワークがダウンした後、特定のネットワーク リンクが他のリンクより容易に、または短時間に修復される場合もあります。信頼性のランクを割り当てるときに考慮できる信頼性係数は、一般的にネットワーク リンクに割り当てる任意の数値です。

ルーティング遅延

ルーティング遅延は、 送信元から宛先に、インターネットワークを通過してパケットを移動するために必要な時間の長さです。遅延は、中間のネットワーク リンクの帯域幅、経由する各ルータでのポートキュー、中間の全ネットワーク リンクでのネットワークの輻輳状況、パケットが移動する物理的な距離など、多くの要素に応じて異なります。ルーティング遅延はいくつかの重要な変数の組み合わせであるため、一般的で便利なメトリックです。

帯域幅

帯域幅は、リンクで使用可能なトラフィック容量です。たとえば、10 ギガビット イーサネット リンクは1 ギガビット イーサネット リンクより優れています。帯域幅は、リンクで達成可能な最大スループットですが、帯域幅のより大きいリンクを経由するルートが、帯域幅のより小さいリンクを経由するルートより優れているとは限りません。たとえば、帯域幅の大きいリンクの方が混雑していると、実際には、パケットを宛先に送信するためにさらに長い時間がかかる場合があります。

負荷

負荷は、ルータなどのネットワーク リソースの使用状況の度合いです。負荷は、CPU 使用状況や処理される 1 秒あたりのパケット数など、さまざまな方法で計算できます。これらのパラメータを継続的にモニタすると、リソースに負担がかかる場合があります。

通信コスト

通信コストは、リンク上でルーティングするための稼働コストの測定単位です。通信コストは重要なメトリックの 1 つで、特にパフォーマンスより稼働コストの削減が優先される場合に使用されます。たとえば、専用回線での回線遅延が公衆回線より大きくても、使用時間に応じて課金される公衆回線上でなく、自身の専用回線上でパケットを送信できます。

ルータ ID

各ルーティング プロセスには、ルータ ID が関連付けられています。ルータ ID は、システムのあらゆるインターフェイスに設定できます。ルータ ID を設定しないと、Cisco NX-OS が次の基準に基づいて、ルータ ID を選択します。

- Cisco NX-OS は、他のあらゆるインターフェイス上で loopback0 を優先します。loopback0 が存在しない場合、Cisco NX-OS は、他のあらゆるインターフェイス タイプ上で最初のループバックを優先します。
- ループバック インターフェイスを設定しなかった場合、Cisco NX-OS はルータ ID としてコンフィギュレーション ファイルの最初のインターフェイスを使用します。Cisco NX-OS がルータ ID を選択した後にいずれかのループバック インターフェイスを設定した場合は、ループバック インターフェイスがルータ ID となります。ループバック インターフェイスが loopback0 ではなく、loopback0 を IP アドレスで設定した場合は、ルータ ID が loopback0 の IP アドレスに変更されます。
- ルータ ID の元であるインターフェイスが変更されると、新しい IP アドレスがルータ ID となります。他のどのインターフェイスの IP アドレスが変更されても、ルータ ID はまったく変更されません。

自律システム

自律システム（AS）とは、単一の技術的管理エンティティにより制御されるネットワークです。自律システムにより、グローバルな外部ネットワークが個々のルーティング ドメインに分割され、これらのドメインでは、ローカルのルーティング ポリシーが適用されます。この構成により、ルーティング ドメインの管理と一貫したポリシー設定が簡素化されます。

各自律システムは、ルートの再配布により動的にルーティング情報を交換する、複数の内部ルーティング プロトコルをサポートできます。地域インターネット レジストリ（RIR）により、インターネットに直接接続する各公共 AS に一意の番号が割り当てられます。この自律システム番号で、ルーティング処理と自律システムの両方が識別されます。

ボーダー ゲートウェイ プロトコル（BGP）は、asplain と asdot 表記で表示できる 4 バイトの AS 番号をサポートします。

- asplain：10 進表記方式。2 バイトおよび 4 バイト AS 番号をその 10 進数値で表します。たとえば、65526 は 2 バイト AS 番号、234567 は 4 バイト AS 番号になります。

- asdot：AS ドット付き表記方式。2 バイト AS 番号をその 10 進数値で表し、4 バイトの AS 番号をドット付き表記で表します。たとえば、2 バイト AS 番号 65526 は 65526 として表され、4 バイトの AS 番号 65546 は 1.10 として表されます。

BGP の 4 バイト AS 番号機能は、4 バイト AS 番号をサポートしていない BGP スピーカーをまたがって、4 バイトをベースとする AS パス情報を伝播するために使用されます。

（注） RFC 5396 は部分的にサポートされます。asplain と asdot 表記はサポートされますが、asdot+ 表記はサポートされません。

専用自律システム番号は内部ルーティング ドメインに使用されますが、インターネット上にルーティングされたトラフィック向けに、ルータにより変換される必要があります。ルーティング プロトコルを、専用自律システム番号が外部ネットワークにアドバタイズされるように設定しないでください。デフォルトでは、Cisco NX-OS は専用自律システム番号をルーティング更新情報から削除しません。

（注） 公共ネットワークおよび専用ネットワークの自律システム番号は、インターネット割り当て番号局（IANA）により管理されています。予約済み番号の割り当てを含む自律システム番号の詳細について、または、AS番号の登録を申請するには、次のURLを参照してください：http://www.iana.org/

コンバージェンス

ルーティング アルゴリズム測定の鍵となる要素の 1 つは、ルータがネットワーク トポロジの変化に対応するために要する時間です。リンク障害など、なんらかの理由でネットワークの一部が変化すると、さまざまなルータのルーティング情報が一致しなくなる場合があります。変化したトポロジに関する情報が更新されているルータと、古い情報が残っているルータがあるためです。コンバージェンスとは、ネットワーク内のすべてのルータが更新され、ルーティング情報が一致するまでにかかる時間の長さです。コンバージェンス時間は、ルーティング アルゴリズムによって異なります。コンバージェンスが速い場合は、不正確なルーティング情報によるパッケージ損失の可能性が小さくなります。

ロード バランシングおよび等コスト マルチパス

ルーティングプロトコルは、ロードバランシングまたは等コストマルチパス（ECMP）を使用して、複数のパス間でトラフィックを共有できます。ルータは、特定のネットワークへの複数のルートを認識すると、最短のアドミニストレーティブディスタンスを持つルートをルーティングテーブルにインストールします。ルータが、同じアドミニストレーティブ ディスタンスと宛先までのコストを持つ複数のパスを受信し、インストールすると、ロード バランシングが発生する場合があります。ロード バランシングでは、すべてのパス上にトラフィックが配布され、負荷が共有されます。使用されるパスの数は、ルーティングプロトコルによりルーティングテーブルに配置されるエントリの数に制限されます。各ルーティングプロトコルによってサポートされている ECMP

の数については、『Cisco Nexus 9000 シリーズ NX-OS 検証済みのスケーラビリティ ガイド』を参照してください。

（注） ECMP は、すべてのリンクで均等なロードバランシングを保証するわけではありません。特定のフローが任意の時点で 1 つの特定のネクスト ホップを選択することだけを保証します。

ルートの再配布の概要

ネットワークに複数のルーティング プロトコルが設定されている場合は、各プロトコルにルートの再配布を設定して、ルーティング情報を共有するように設定できます。たとえば、OSPF（Open Shortest Path First）プロトコルを設定して、ボーダー ゲートウェイ プロトコル（BGP）で検出したルートをアドバタイズできます。また、スタティック ルートを、どのダイナミック ルーティング プロトコルにも再配布できます。他のプロトコルからのルートを再配布するルータは、異なるルーティング プロトコル間で互換性のないルート メトリックを防ぐ再配布されたルータの固定ルートを設定します。たとえば、EIGRP から OSPF に再配布されたルートには、OSPF が認識できる固定リンク コスト メトリックが割り当てられます。

（注） ルーティング情報の再配布を設定する場合にルート マップを使用する必要があります。

ルート再配布では、アドミニストレーティブ ディスタンス（「アドミニストレーティブ ディスタンス」セクションを参照）の使用によっても、2 つの異なるルーティング プロトコルで検出されたルートが区別されます。優先ルーティング プロトコルには、より低いアドミニストレーティブ ディスタンスが与えられており、そのルートが、より高いアドミニストレーティブ ディスタンスが割り当てられた他のプロトコルからのルートに優先して選択されます。

アドミニストレーティブ ディスタンス

アドミニストレーティブ ディスタンスは、ルーティング情報源の信頼性を示す評価基準です。値が高いほど信頼性の評価は低くなります。一般的にルートは、複数のプロトコルを通じて検出されます。アドミニストレーティブ ディスタンスは、複数のプロトコルから学習したルートを区別するために使用されます。最もアドミニストレーティブ ディスタンスが低いルートが IP ルーティング テーブルに組み込まれます。

スタブ ルーティング

スタブ ルーティングはハブ アンド スポーク型ネットワーク トポロジで使用できます。このトポロジでは、1 つ以上の終端（スタブ）ネットワークが 1 台のリモートルータ（スポーク）に接続され、そのリモート ルータは 1 つ以上のディストリビューション ルータ（ハブ）に接続されています。リモート ルータは、1 つ以上のディストリビューション ルータにのみ隣接しています。リモート ルータへ流れる IP トラフィックのルートは、ディストリビューション ルータ経由のルートのみです。このタイプの設定は、ディストリビューション ルータが直接 WAN に接続されてい

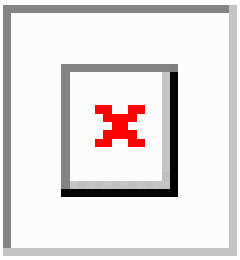
る WAN トポロジで使用されるのが一般的です。ディストリビューション ルータは、さらに多くのリモート ルータに接続できます。ディストリビューション ルータが 100 台以上のリモート ルータに接続されていることも、よくあります。ハブ アンド スポーク型トポロジでは、リモート ルータがすべての非ローカル トラフィックをディストリビューション ルータに転送する必要があります。これにより、リモート ルータが完全なルーティング テーブルを保持する必要はなくなります。通常、分散ルータは、デフォルトのルートのみをリモート ルータに送信します。

指定されたルートのみが、リモート（スタブ）ルータから伝播されます。スタブ ルータは、サマリー、接続されているルート、再配布されたスタティック ルート、外部ルート、および内部ルートに対するクエリーすべてに、応答として「inaccessible」というメッセージを返します。スタブとして設定されているルータは、自身のスタブ ルータとしてのステータスを報告するために、特殊なピア情報パケットがすべての隣接ルータに送信されます。

スタブ ルータの状態を通知するパケットを受信した隣接ルータは、ルートについてはスタブ ルータに照会しません。また、スタブ ピアを持つルータは、そのピアについては照会しません。スタブ ルータは、ディストリビューション ルータを使用して適切なアップデートをすべてのピアに送信します。

次の図は、単純なハブ アンド スポーク型のコンフィギュレーションを示しています。

図 *2:* 単純なハブ アンド スポーク ネットワーク



スタブ ルーティングを使用する場合でも、リモート ルータにルータをアドバタイズできます。この**単純なハブ アンド スポーク ネットワーク**の図は、リモート ルータが、分散ルータを介してのみ、企業ネットワークとインターネットにアクセスできることを示しています。この例では、企業ネットワークとインターネットへのパスが常に分散ルータを経由するため、リモート ルータ上の完全なルート テーブルの機能は無意味です。より大規模なルート テーブルを使用しても、リモート ルータに必要なメモリの量が削減されるだけです。使用される帯域幅とメモリは、分散ルータでルートを要約し、フィルタリングすると、削減できます。このネットワーク トポロジでリモート ルータは、他のネットワークから検出されたルートを受信する必要はありません。これは、宛先がどこであっても、リモート ルータは、すべての非ローカル トラフィックを分散ルータに送信する必要があるためです。真のスタブ ネットワークを設定するには、リモート ルータへのデフォルト ルートのみを送信するよう、分散ルータを設定する必要があります。

OSPF はスタブ エリアをサポートして、Enhanced Interior Gateway Routing Protocol（EIGRP）はスタブ ルータをサポートします。

（注） EIGRP スタブ ルーティング機能は、スタブ デバイスだけで使用します。スタブ デバイスは、コア中継トラフィックが通過しないネットワーク コアまたはディストリビューション レイヤに接続されたデバイスとして定義されます。リモートルータへ流れる IP トラフィックのルートは、ディストリビューション ルータ経由のルートのみです。スタブ デバイスがディストリビューション デバイス以外の EIGRP ネイバーを持つことはできません。この制限を無視すると、望ましくない動作が発生します。

ルーティング アルゴリズム

ルーティング アルゴリズムによって、ルータが到達可能性情報を収集して報告する方法、トポロジの変化に対応する方法、宛先までの最適ルートを決定する方法が決まります。ルーティング アルゴリズムにはさまざまなタイプがあり、各アルゴリズムがネットワークやルータ リソースに与える影響もさまざまです。ルーティング アルゴリズムは、最適なルートの計算に影響するさまざまなメトリックを使用します。ルーティング アルゴリズムは、スタティックまたはダイナミック、内部または外部など、タイプで分類できます。

スタティック ルートおよびダイナミック ルーティング プロトコル

スタティック ルートは、手動で設定するルートテーブルエントリです。スタティック ルートは、手動で再設定しない限り、変更されません。スタティック ルートは設計が簡単で、ネットワーク トラフィックが比較的予想しやすい環境や、ネットワーク設計が比較的単純な環境での使用に適しています。

スタティック ルーティング システムはネットワークの変化に対応できないため、絶えず変化する大規模ネットワークには使用しないでください。今日のほとんどのルーティング プロトコルは、ダイナミック ルーティング アルゴリズムを使用しています。このアルゴリズムでは、着信ルーティング更新メッセージを分析して、ネットワーク状況の変化に合わせて調整します。メッセージがネットワークが変化したことを示している場合は、ルーティング ソフトウェアはルートを再計算し、新しいルーティングアップデートメッセージを送信します。これらのメッセージがネットワークを通過すると、ルータがそのアルゴリズムを再実行し、それに従ってルーティング テーブルを変更します。

適切であれば、ダイナミック ルーティング アルゴリズムをスタティック ルートで補完することができます。たとえば、各サブネットワークに IP デフォルト ゲートウェイまたは、ラストリゾートルータ（ルーティングできないすべてのパケットが送信されるルータ）へのスタティック ルートを設定する必要があります。

内部および外部ゲートウェイ プロトコル

ネットワークを、一意のルーティング ドメインまたは自律システムに分割できます。自律システムは、管理ガイドラインの特定のセットで規制された共通の管理機関の下の内部ネットワークの一部です。自律システム間でのルートを設定するルーティング プロトコルは、外部ゲートウェイ

プロトコルまたはドメイン間プロトコルと呼ばれます。ボーダーゲートウェイプロトコル（BGP）は、外部ゲートウェイ プロトコルの例です。1 つの自律システム内で使用されるルーティング プロトコルは、内部ゲートウェイ プロトコルまたはドメイン内プロトコルと呼ばれます。EIGRP および OSPF は、内部ゲートウェイ プロトコルの例です。

ディスタンス ベクトル プロトコル

ディスタンス ベクトル プロトコルは、ディスタンス ベクトル アルゴリズム（Bellman-Ford アルゴリズムとも呼ばれます）を使用します。このアルゴリズムにより、各ルータは、そのルーティング テーブルの一部または全部を隣接ルータに送信します。ディスタンス ベクトル アルゴリズムでは、ルートが、ディスタンス（宛先までのホップ数など）および方向（ネクストホップ ルータなど）により定義されます。その後、これらのルートは、直接接続されたネイバー ルータにブロードキャストされます。各ルータは、これらの更新情報を使用して、ルーティング テーブルを確認し、更新します。

ルーティング ループを防ぐために、ほとんどのディスタンス ベクトル アルゴリズムはポイズン リバースを指定したスプリット ホライズンを使用します。これは、インターフェイスで検出されたルートを到達不能として設定し、それをそのインターフェイスで、次の定期更新中にアドバタイズするという意味です。このプロセスにより、ルータによるルート更新が、そのルータ自体に返信されなくなります。

ディスタンス ベクトル アルゴリズムは、一定の間隔で更新を送信しますが、ルート メトリックの値の変更に応じて、更新を送信することもできます。このように送信された更新により、ルート コンバージェンス時間の短縮が可能です。Routing Information Protocol（RIP）はディスタンス ベクトル プロトコルの 1 つです。

リンクステート プロトコル

リンク ステート プロトコルは、最短パス優先（SPF）とも呼ばれ、情報を隣接ルータと共有します。各ルータは、各リンクおよび直接接続されたネイバー ルータに関する情報を含むリンク ステート アドバタイズメント（LSA）を構築します。

各 LSA にはシーケンス番号があります。ルータが LSA を受信し、そのリンクステート データベースを更新すると、その LSA はすべての隣接ネイバーにフラッディングされます。ルータが（同じルータから）同じシーケンス番号の 2 つの LSA を受信した場合、ルータは LSA アップデートのループを回避するため、ネイバーによって受信された最後の LSA をフラッディングしません。ルータは、受信直後に LSA をフラッディングするため、リンクステート プロトコルのコンバージェンス時間は最小となります。

ネイバー ルータの探索と隣接関係の確立は、リンクステートプロトコルの重要な部分です。ネイバー ルータは、特別な hello パケットを使用して探索されます。このパケットは、各ネイバー ルータのキープアライブ通知としても機能します。隣接関係は、ネイバー ルータ間のリンクステート プロトコルの一般的な動作パラメータ セットで確立されます。

ルータが受信した LSA は、そのルータのリンクステート データベースに追加されます。各エントリは、次のパラメータで構成されます。

- ルータ ID（LSA を構築したルータの）

- ネイバー ID
- リンク コスト
- LSA のシーケンス番号
- LSA エントリの作成時からの経過時間

ルータは、リンクステート データベース上で SPF アルゴリズムを実行し、そのルータの最短パス ツリーを構築します。この SPF ツリーを使用して、ルーティング テーブルにデータが入力されます。

リンクステートアルゴリズムでは、各ルータはネットワークの全体像をそのルーティングテーブルに構築します。リンクステート アルゴリズムが小さな更新を全体的に送信するのに対し、ディスタンス ベクトル アルゴリズムは、より大きな更新をネイバー ルータのみに送信します。

リンクステートアルゴリズムは、より短時間でコンバージェンスするため、ディスタンスベクトル アルゴリズムより、ルーティング ループがやや発生しにくくなっています。ただし、リンクステート アルゴリズムは、ディスタンス ベクトル アルゴリズムより、より多くの CPU パワーとメモリを必要とし、実行とサポートをするにはよりコストが高くなります。一般的に、リンクステート プロトコルはディスタンス ベクトル プロトコルよりもスケーラブルです。

OSPF は、リンクステート プロトコルの一例です。

レイヤ 3 仮想化

Cisco NX-OS は、複数の仮想ルーティングおよび転送（VRF）インスタンスおよび複数のルーティング情報ベース（RIB）をサポートしているため、複数のアドレス ドメインがサポートされます。各 VRF は RIB に関連付けられており、この情報が転送情報ベース（FIB）によって収集されます。VRF は、レイヤ 3 アドレス指定ドメインを表します。各レイヤ 3 インターフェイス（論理または物理）は、1 つの VRF に属します。詳細については、「レイヤ 3 仮想化の設定」を参照してください。

Cisco NX-OS では、仮想デバイスをエミュレートする Virtual Device Context（VDCs）に、OS およびハードウェア リソースを分割できます。Cisco Nexus 9000 シリーズ スイッチは、現在のところ、複数の VDC をサポートしていません。すべてのスイッチ リソースはデフォルト VDC で管理されます。

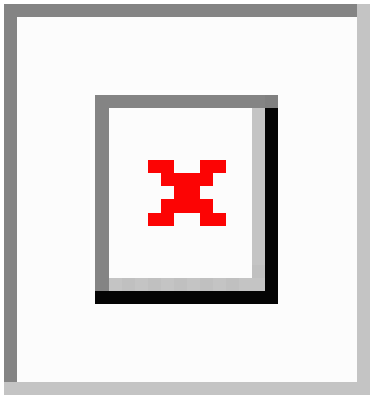
Cisco NX-OS フォワーディング アーキテクチャ

Cisco NX-OS では、転送アーキテクチャにより、すべてのルーティングの更新処理と、シャーシ内のすべてのモジュールへの転送情報の入力が行われます。

ユニキャスト RIB

Cisco NX-OS 転送アーキテクチャは、次の図に示すように、複数のコンポーネントから構成されています。

図 *3 : Cisco NX-OS* 転送アーキテクチャ



ユニキャスト RIB はアクティブなスーパーバイザ上にあります。ユニキャスト RIB は、直接接続のルート、スタティック ルート、ダイナミック ユニキャスト ルーティング プロトコルで検出されたルートを含むルーティング テーブルを維持しています。また、アドレス解決プロトコル（ARP）などの送信元から、隣接情報を収集します。ユニキャスト RIB は、特定のルートのための最適なネクストホップを決定し、ユニキャスト FIB 分散モジュール（FDM）のサービスを使用して、FIB にデータを入力します。

各ダイナミック ルーティング プロトコルは、タイム アウトしたあらゆるルートについて、ユニキャスト RIB を更新する必要があります。その後、ユニキャスト RIB はそのルートを削除し、そのルートに最適なネクスト ホップを再計算します（代わりに使用できるパスがある場合）。

隣接マネージャ

隣接マネージャはアクティブなスーパーバイザ上にあり、ARP、ネイバー探索プロトコル（NDP）、スタティック設定など、各種プロトコルの隣接情報を保持しています。最も基本的な隣接情報は、これらのプロトコルで探索されたレイヤ 3 からレイヤ 2 へのアドレス マッピングです。発信レイヤ 2 パケットは、隣接情報を使用して、レイヤ 2 ヘッダーの作成を終了します。

隣接マネージャは、ARP 要求による、レイヤ 3 からレイヤ 2 への特定のマッピングの探索をトリガーできます。新しいマッピングは、対応する ARP 返信を受信し、処理すると、使用できるようになります。IPv6 の場合は、隣接マネージャが NDP からの、レイヤ 3 からレイヤ 2 へのマッピング情報を探索します。詳細については、IPv6 の設定 （59 ページ） を参照してください。

ユニキャスト転送分散モジュール

ユニキャスト転送分散モジュール（FDM）はアクティブなスーパーバイザ上に存在し、ユニキャスト RIB やその他の送信元からの転送パス情報を配布します。ユニキャスト RIB は、ユニキャスト FIB によってスタンバイ スーパーバイザおよびモジュール上のハードウェア転送テーブルにプ

ログラミングされる転送情報を生成します。また、ユニキャストFDMは、新規挿入されたモジュールへの FIB 情報のダウンロードも行います。

ユニキャスト FDM は隣接関係情報を収集し、ユニキャスト FIB でのルート更新時に、この情報およびその他のプラットフォーム依存の情報を書き直し（リライトし）ます。隣接情報およびリライト情報には、インターフェイス、ネクスト ホップ、およびレイヤ 3 からレイヤ 2 へのマッピング情報が含まれています。インターフェイスとネクストホップの情報は、ユニキャスト RIB からのルート更新情報で受信します。レイヤ 3 からレイヤ 2 へのマッピングは、隣接マネージャから受信します。

FIB

ユニキャスト FIB は、スーパーバイザ モジュールとスイッチング モジュール上にあり、ハードウェア転送エンジンで使用される情報を構築します。ユニキャスト FIB は、ユニキャスト FDM からルート更新情報を受信し、ハードウェア転送エンジンにプログラミングされるよう、この情報を送信します。ユニキャスト FIB は、ルート、パス、隣接関係の追加、削除、変更を管理します。

ユニキャスト FIB は VRF 単位および address-family 単位で保持されます。つまり、設定された各 VRF に対して IPv4 用に 1 つ、IPv6 用に 1 つが保持されます。ルート更新メッセージに基づいて、ユニキャスト FIB は、VRF ごとのプレフィックスとネクストホップ隣接情報データベースを維持します。ネクストホップ隣接データ構造には、ネクストホップの IP アドレスとレイヤ 2 リライト情報が含まれます。同じネクストホップ隣接情報構造を複数のプレフィックスで使用できます。

ハードウェア フォワーディング

Cisco NX-OS は、分散パケット転送をサポートします。入力ポートは、パケット ヘッダーから該当する情報を取得し、その情報をローカル スイッチング エンジンに渡します。ローカル スイッチング エンジンはレイヤ 3 ルックアップを行い、この情報を使って、パケット ヘッダーをリライトします。入力モジュールは、パケットを出力ポートに転送します。出力ポートが別のモジュール上にある場合は、スイッチ ファブリックを使って、パケットが出力モジュールに転送されます。出力モジュールは、レイヤ 3 転送決定には関与しません。

また、 **show platform fib**、 または **show platform forwarding** コマンドを使用して、ハードウェア転送の詳細を表示することもできます。

ソフトウェア転送

Cisco NX-OS のソフトウェア転送パスは、主に、ハードウェアでサポートされない機能、またはハードウェア処理中に発生したエラーへの対処に使用されます。通常、IP オプション付きのパケットまたはフラグメンテーションの必要なパケットは、アクティブなスーパーバイザ上の CPU に渡されます。ソフトウェアでの切り替えが必要なパケットや終端される必要のあるパケットはすべて、スーパーバイザに渡されます。スーパーバイザは、ユニキャスト RIB および隣接マネージャから提供された情報を使用して、転送の決定を下します。モジュールは、ソフトウェア転送パスには関与しません。

ソフトウェア転送は、コントロール プレーン ポリシーおよびレート リミッタによって管理されます。詳細については、「Cisco NX-OS 9000 シリーズ NX-OS セキュリティ設定ガイド」を参照してください。

レイヤ 3 ユニキャスト ルーティング機能のまとめ

ここでは、Cisco NX-OS でサポートされるレイヤ 3 ユニキャスト機能およびプロトコルを簡単に説明します。

IPv4 and IPv6

レイヤ 3 は、IPv4 プロトコルまたは IPv6 プロトコルを使用します。IPv6 では、ネットワーク アドレス ビット数が 32 ビット（IPv4 の場合）から 128 ビットに増やされています。詳細については、IPv4 の設定 （19 ページ）またはIPv6 の設定 （59 ページ）を参照してください。

IP サービス

IP サービスには、DHCP クライアントおよびドメイン ネーム システム（DNS）クライアントがあります。詳細については、「DNS の設定」を参照してください。

Open Shortest Path First（OSPF）

Open Shortest Path First（OSPF）プロトコルは、AS 内のネットワーク到達可能性情報の交換に使用されるリンクステート ルーティング プロトコルです。各 OSPF ルータは、そのアクティブなリンクに関する情報をネイバー ルータにアドバタイズします。リンク情報には、リンク タイプ、リンク メトリック、およびリンクに接続された隣接ルータが含まれます。このリンク情報を含むアドバタイズメントは、リンクステートアドバタイズメントと呼ばれます。詳細については、OSPFv2 の設定 （113 ページ） を参照してください。

EIGRP

nhanced Interior Gateway Routing Protocol（EIGRP）は、ディスタンス ベクトルとリンクステートの両ルーティングプロトコルの特徴を備えたユニキャストルーティングプロトコルです。これは、シスコ専用ルーティング プロトコルである IGRP の改良バージョンです。EIGRP はネイバーに依存し、ルートを提供します。また、リンクステートプロトコルのように、ネイバー ルータからアドバタイズされたルートからネットワーク トポロジを構築し、この情報を使用して、ループの発生しない、宛先までのパスを選択します。詳細については、EIGRP の設定 （237 ページ） を参照してください。

IS-IS

Intermediate System-to-Intermediate System（IS-IS）プロトコルは、国際標準化機構（ISO）10589 で指定されたドメイン内開放型システム間相互接続（Open System Interconnection）ダイナミック ルー

ティングプロトコルです。IS-IS ルーティングプロトコルはリンクステートプロトコルです。IS-IS 機能は次のとおりです。

- 階層型ルーティング
- クラスレス動作
- 新情報の高速フラッディング
- 短時間でのコンバージェンス
- 高いスケーラビリティ

詳細については、IS-IS の設定 （279 ページ） を参照してください。

BGP

BGP は自律システム間ルーティングプロトコルです。BGP ルータは、信頼性の高い転送メカニズムとして伝送制御プロトコル（TCP）を使用し、他の BGP ルータにネットワーク到達可能性情報をアドバタイズします。ネットワーク到達可能性情報には、宛先ネットワーク プレフィックス、宛先に到達するまでに通過する必要のある自律システムのリスト、およびネクストホップ ルータが含まれます。到達可能性情報には、ルートの優先度、ルートの始点、コミュニティなどの詳細なパス属性が含まれます。詳細については、「基本的 BGP の設定 （319 ページ）」および「高度な BGP の設定 （357 ページ）」を参照してください。

RIP

RIP は、ホップ数をメトリックとして使用するディスタンス ベクトル プロトコルです。RIP は、世界中のインターネットでトラフィックのルーティングに広く使用されています。また、IGP であるため、単一の自律システム内でルーティングを行います。詳細については、RIP の設定（487 ページ） を参照してください。

スタティック ルーティング

スタティック ルーティングを使用して、宛先までの一定のルートを入力できます。この機能は、単純なトポロジの小規模ネットワークでは便利です。また、スタティック ルーティングは、他のルーティングプロトコルとともに、デフォルトルートおよびルート配布の管理に使用されます。詳細については、「スタティック ルーティングの設定」を参照してください。

レイヤ 3 仮想化

仮想化を使用すると、複数の管理ドメインにわたる物理リソースを共有できます。Cisco NX-OS は、仮想ルーティングおよび転送（VRF）を含むレイヤ 3 仮想化をサポートしています。VRF では、レイヤ 3 ルーティング プロトコルを設定するための別のアドレス ドメインが提供されます。詳細については、「レイヤ 3 仮想化の設定」を参照してください。

Route Policy Manager

Route Policy Manager は、でルート フィルタリング機能を提供します。Route Policy Manager はルート マップを使用して、さまざまなルーティング プロトコルや、特定のルーティング プロトコル内のさまざまなエンティティ間で配布されたルートをフィルタリングします。フィルタリングは、特定の一致基準に基づいて行われます。これは、アクセス コントロール リストによるパケット フィルタリングに似ています。詳細については、Route Policy Manager の設定 （587 ページ） を参照してください。

ポリシーベース ルーティング

ポリシーベース ルーティングは、Route Policy Manager を使用してポリシー ルート フィルタを作成します。これらのポリシー ルート フィルタでは、パケットの送信元またはパケット ヘッダーのその他フィールドに基づいて、指定されたネクスト ホップにパケットを転送できます。プロトコル タイプやポート番号に基づいてルーティングできるように、ポリシー ルートを拡張 IP アクセスリストにリンクすることができます。詳細については、「ポリシーベーステンプレートの設定」を参照してください。

ファーストホップ冗長プロトコル（FHRP）

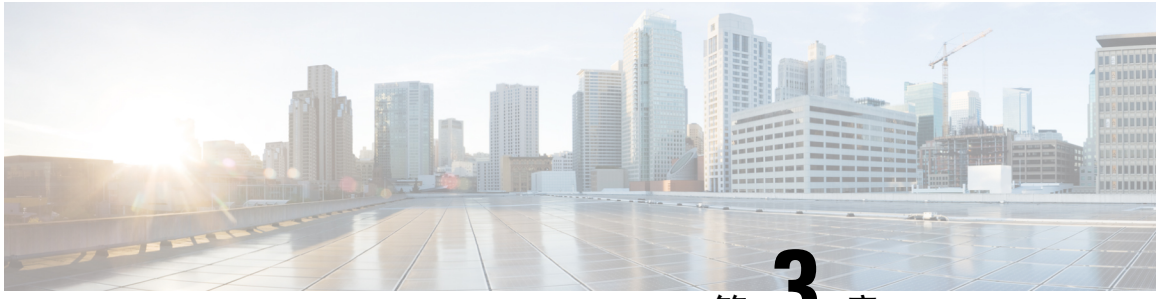
ホットスタンバイルータプロトコル（HSRP）、仮想ルータ冗長プロトコル（VRRP）などのファースト ホップ冗長プロトコル（FHRP）を使用すると、ホストで接続の冗長性を実現できます。アクティブなファーストホップルータがダウンした場合は、その機能を引き継ぐスタンバイルータが FHRP によって自動的に選択されます。アドレスは仮想のものであり、FHRP グループ内の各ルータ間で共有されているため、ホストを新しいIPアドレスで更新する必要はありません。HSRP の詳細については、「HSRP の設定」を参照してください。VRRPの詳細については、VRRP の設定 （681 ページ） を参照してください。

オブジェクト トラッキング

オブジェクト トラッキングを使用すると、インターフェイス回線プロトコル状態、IP ルーティング、ルート到達可能性などの、ネットワーク上の特定のオブジェクトをトラッキングし、トラッキングしたオブジェクトの状態が変化したときに対処することができます。この機能により、ネットワークのアベイラビリティが向上し、オブジェクトがダウンした場合のリカバリ時間が短縮されます。詳細については、「オブジェクト トラッキングの設定」を参照してください。

関連項目

機能名	機能情報
レイヤ 3 機能	「*Cisco NX-OS 9000* シリーズ *NX-OS* マルチキャスト ルーティング設定ガイド」 「*Cisco Cisco NX-OS 9000* シリーズ *NX-OS* 高可用性および冗長性ガイド」 自律システムの数を検索する:https://www.iana.org/numbers



第 3 章

IPv4 の設定

この章では、Cisco NX-OS デバイス上でのインターネット プロトコル バージョン 4（IPv4）（アドレス指定を含む）、アドレス解決プロトコル（ARP）および Internet Control Message Protocol（ICMP）の設定方法を説明します。

この章は、次の項で構成されています。

- IPv4 の概要 （19 ページ）
- IPv4 の仮想化のサポート （28 ページ）
- IPv4の前提条件 （28 ページ）
- IPv4 の注意事項および制約事項 （28 ページ）
- デフォルト設定 （30 ページ）
- IPv4 の設定 （31 ページ）
- IPv4 設定の確認 （56 ページ）
- その他の参考資料 （57 ページ）

IPv4 の概要

デバイス上で IP を設定し、ネットワーク インターフェイスに IP アドレスを割り当てることができます。IP アドレスを割り当てると、インターフェイスがイネーブルになり、そのインターフェイス上のホストと通信できるようになります。

IP アドレスは、デバイス上でプライマリまたはセカンダリとして設定できます。インターフェイスには、1 つのプライマリ IP アドレスと複数のセカンダリ アドレスを設定できます。デバイスが生成したパケットは、常にプライマリ IPv4 アドレスを使用するため、インターフェイス上のすべてのネットワーキング デバイスは、同じプライマリ IP アドレスを共有する必要があります。各 IPv4 パケットは、送信元または宛先 IP アドレスからの情報に基づいています。詳細については、「複数 IP アドレス」の項を参照してください。

サブネットを使用して、IP アドレスをマスクできます。マスクは、IP アドレスがどのサブネットに属するかを決定するために使用されます。IPアドレスは、ネットワークアドレスとホストアドレスで構成されています。マスクで、IP アドレス中のネットワーク番号を示すビットが識別できます。マスクを使用してネットワークをサブネット化した場合、そのマスクはサブネット マスク

と呼ばれます。サブネット マスクは 32 ビット値で、これにより IP パケットの受信者は、IP アドレスのネットワーク ID 部分とホスト ID 部分を区別できます。

IP 機能には、スーパーバイザ モジュールで終端する IPv4 パケットを取り扱い、また同様に、IPv4 ユニキャスト/マルチキャスト ルート ルックアップとソフトウェア アクセス コントロール リスト（ACL）の転送を含む IPv4 パケットの転送を行う役割があります。また、IP 機能は、ネットワーク インターフェイス IP アドレス設定、重複アドレス チェック、スタティック ルート、および IP クライアントのパケット送信/受信インターフェイスも管理します。

（注） Nexusの動作ではnull0インターフェイス宛てのパケットがドロップされるため、IPv4またはIPv6パケットがnull0インターフェイスに送信された場合、Cisco Nexus 3000スイッチはICMPまたはICMPv6パケットで応答しません。

複数の IPv4 アドレス

Cisco NX-OS は、インターフェイスごとに複数の IP アドレスをサポートします。さまざまな状況に備え、いくつでもセカンダリ アドレスを指定できます。最も一般的な状況は次のとおりです。

- 特定のネットワーク インターフェイスのホスト IP アドレスの数が不足している場合。たとえば、サブネット化により、論理サブネットごとに 254 までのホストを使用できるが、物理サブネットの 1 つに 300 のホスト アドレスが必要な場合は、ルータ上またはアクセス サーバ上でセカンダリ IP アドレスを使用して、1 つの物理サブネットで 2 つの論理サブネットを使用できます。
- 1 つのネットワークの 2 つのサブネットは、別の方法で、別のネットワークにより分離できる場合があります。別のネットワークによって物理的に分離された複数のサブネットから、セカンダリ アドレスを使用して、1 つのネットワークを作成できます。このような場合、最初のネットワークは、2 番めのネットワークの上に拡張されます。つまり、上の階層となります。サブネットは、同時に複数のアクティブなインターフェイス上に表示できません。

（注） ネットワーク セグメント上のいずれかのデバイスがセカンダリ IPv4 アドレスを使用している場合は、同じネットワーク インターフェイス上の他のすべてのデバイスも、同じネットワークまたはサブネットからのセカンダリ アドレスを使用する必要があります。ネットワーク セグメント上で、一貫性のない方法でセカンダリ アドレスを使用すると、ただちにルーティング ループが発生する可能性があります。

LPMルーティングモード

デフォルトでは、Cisco NX-OSは、デバイス上で最長プレフィックス一致（LPM）を許可するように階層的にルーティングします。ただし、より多くの LPM ルート エントリをサポートするために、異なるルーティング モード用にデバイスを設定できます。

次の表に、Cisco Nexus 9300 シリーズおよび 9500 シリーズ スイッチでサポートされている LPM ルーティング モードを示します。

表 *2 : Cisco Nexus 9200* シリーズ スイッチ用の *LPM* ルーティング モード

LPM ルーティング モード	CLI コマンド
デフォルトのシステム ルーティング モード	
LPM デュアルホスト ルーティング モード	**system routing template-dual-stack-host-scale**
LPM ヘビー ルーティング モード	**system routing template-lpm-heavy**

（注） Cisco Nexus 9200 プラットフォーム スイッチは、IPv4 マルチキャスト ルートの **system routing template-lpm-heavy** モードをサポートしていません。LPM の上限を 0 にリセットしてください。

表 *3 : Cisco Nexus 9300* シリーズ スイッチ用の *LPM* ルーティング モード

LPM ルーティング モード	Broadcom T2モード	CLI コマンド
デフォルトのシステム ルーティング モード	3	
ALPM ルーティング モード	4	**system routing max-mode l3**

表 *4 : Cisco Nexus 9300-EX/FX/FX2/FX3/GX* プラットフォーム スイッチ用の *LPM* ルーティング モード

LPM ルーティング モード	CLI コマンド
LPM デュアルホスト ルーティング モード	**system routing template-dual-stack-host-scale**
LPM ヘビー ルーティング モード	**system routing template-lpm-heavy**
LPM インターネットピアリング モード)	**system routing template-internet-peering**

表 *5 : 9700-EX* および *9700-FX* ラインカードを搭載した *Cisco Nexus 9500* プラットフォーム スイッチ用 *LPM* ルーティング モード

LPM ルーティング モード	Broadcom T2モード	CLI コマンド
デフォルトのシステムルーティング モード	3（ラインカード用）。 4（ファブリック モジュール用）	
最大-ホスト ルーティング モード	2（ラインカード用）。 3（ファブリック モジュール用）	**system routing max-mode host**
非階層ルーティング モード	3（ラインカード用）。 max-l3-modeオプション付き4（ラインカード用）	**system routing non-hierarchical-routing [max-l3-mode]**
64 ビット ALPM ルーティング モード	モード4のサブモード（ファブリックモジュール用）	**system routing mode hierarchical 64b-alpm**
LPM ヘビー ルーティング モード		**system routing template-lpm-heavy** （注） このモードは、9732C-EX ライン カードを搭載した Cisco Nexus 9508 スイッチでのみサポートされます。
LPMインターネットピアリング モード)		**system routing template-internet-peering** （注） このモードは、次の Cisco Nexus 9500 プラットフォーム スイッチでのみサポートされています。 • 9700-EX ライン カード搭載の Cisco Nexus 9500 プラットフォーム スイッチ • Cisco Nexus 9500-FX プラットフォーム スイッチ（Cisco NX-OS リリース 7.0(3)I7(4) 以降） • Cisco 9500-R プラットフォーム スイッチ（Cisco NX-OS リリース 9.3(1) 以降）

LPM ルーティング モード	Broadcom T2モード	CLI コマンド
LPM デュアルホスト ルーティング モード		

表 *6 : 9600-R* ライン カードを搭載した *Cisco Nexus 9500-R* プラットフォーム スイッチの *LPM* ルーティング モード

LPM ルーティング モード	CLI コマンド
LPMインターネットピアリングモード)	**system routing template-internet-peering** （Cisco NX-OS リリース 9.3(1) 以降 ）

ホストから LPM へのスピルオーバー

Cisco NX-OS リリース 7.0(3)I5(1) 以降では、ホストルートを LPM テーブルに保存して、より大きなホスト スケールを実現できます。ALPM モードでは、スイッチはより少ないホスト ルートを許可します。サポートされるスケールよりも多くのホストルートを追加すると、ホストテーブルからこぼれたルートは LPM テーブルの LPM ルートのスペースを使用します。このモードで許可される LPM ルートの総数は、保存されているホスト ルートの数だけ減少します。この機能は、Cisco Nexus 9300 および 9300 プラットフォーム スイッチではサポートされていません。

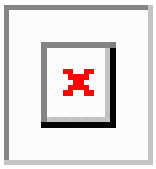
デフォルトのシステム ルーティング モードでは、Cisco Nexus 9300 プラットフォーム スイッチは、より高いホスト スケールとより少ない LPM ルート用に設定され、より多くのホスト ルートを保存するために LPM スペースを使用できます。Cisco Nexus 9500 プラットフォーム スイッチでは、デフォルトのシステムルーティング モードと非階層型ルーティング モードのみがライン カードでこの機能をサポートします。ファブリック モジュールはこの機能をサポートしていません。

アドレス解決プロトコル

ネットワーキング デバイスおよびレイヤ 3 スイッチは ARP を使用して、IP（ネットワーク層）アドレスを物理（Media Access Control（MAC）レイヤ）アドレスにマッピングし、IP パケットがネットワーク上に送信されるようにします。デバイスは、他のデバイスにパケットを送信する前に自身の ARP キャッシュを調べて、MAC アドレスまたは対応する宛先デバイスの IP アドレスがないかを確認します。エントリがまったくない場合、送信元のデバイスは、ネットワーク上の全デバイスにブロードキャスト メッセージを送信します。

各デバイスは、問い合わせられた IP アドレスを自身のアドレスと比較します。一致する IP アドレスを持つデバイスだけが、デバイスの MAC アドレスを含むパケットとともにデータを送信したデバイスに返信します。送信元デバイスは、あとで参照できるよう、宛先デバイスの MAC アドレスをその ARP テーブルに追加し、データリンク ヘッダーおよびトレーラを作成してパケットをカプセル化し、データの転送へと進みます。次の図は、ARP ブロードキャストと応答プロセスを示しています。

図 *4 : ARP* 処理



宛先デバイスが、別のデバイスを挟んだリモート ネットワーク上にあるときは、同じ処理が行われますが、データを送信するデバイスが、デフォルト ゲートウェイの MAC アドレスを求める ARP 要求を送信する点が異なります。アドレスが解決され、デフォルト ゲートウェイがパケットを受信した後に、デフォルト ゲートウェイは、接続されているネットワーク上で宛先の IP アドレスをブロードキャストします。宛先デバイスのネットワーク上のデバイスは、ARP を使用して宛先デバイスの MAC アドレスを取得し、パケットを配信します。ARP はデフォルトでイネーブルにされています。

デフォルトでシステム定義された CoPP ポリシー レートは、スーパーバイザ モジュールにバインドされた ARP ブロードキャスト パケットを制限します。デフォルトのシステム定義 CoPP ポリシーは、ARP ブロードキャスト ストームによるコントロール プレーン トラフィックへの影響を防止し、ブリッジド パケットに影響しません。

ARP キャッシング

ARP キャッシングにより、ブロードキャストが最小になり、ネットワーク リソースの浪費が抑制されます。IP アドレスの MAC アドレスへのマッピングは、ネットワーク間でパケットが送信されるたびに、ネットワーク上の各ホップ（デバイス）で行われるため、ネットワークのパフォーマンスに影響する場合があります。

ARP キャッシングでは、ネットワーク アドレスとそれに関連付けられたデータリンク アドレスが一定の期間メモリ内に保存されるため、パケットが送信されるたびに同じアドレスにブロードキャストするための貴重なネットワーク リソースの使用が最小限に抑えられます。キャッシュ エントリは、定期的に失効するよう設定されているため、保守が必要です。これは、古い情報が無効となる場合があるためです。ネットワーク上のすべてのデバイスは、アドレスのブロードキャストに従ってアドレス テーブルを更新します。

ARP キャッシュのスタティックおよびダイナミック エントリ

スタティック ルーティングは、手動で各デバイスの各インターフェイスに対応する IP アドレス、サブネット マスク、ゲートウェイ、および対応する MAC アドレスを設定する必要があります。スタティック ルーティングでは、ルート テーブルを維持するために、より多くの処理が必要です。ルートを追加または変更するたびに、テーブルの更新が必要となるためです。

ダイナミック ルーティングは、ネットワーク上のデバイスが相互にルーティング テーブル情報を交換できるプロトコルを使用します。ダイナミック ルーティングは、キャッシュに制限時間を追加しない限り、ルートテーブルが自動更新されるため、スタティック ルーティングより効率的です。デフォルトの制限時間は25分ですが、キャッシュから追加および削除されるルートがネットワークに数多く存在する場合は、制限時間を変更します。

ARP を使用しないデバイス

ネットワークが 2 つのセグメントに分割されると、ブリッジによりセグメントが結合され、各セグメントへのトラフィックがMACアドレスに基づいてフィルタリングされます。ブリッジはMACアドレスだけを使用する独自のアドレステーブルを作成します。デバイスがIPアドレスおよび対応する MAC アドレスの両方を含む ARP キャッシュを持っています。

パッシブ ハブは、ネットワーク内の他のデバイスを物理的に接続する集中接続デバイスです。パッシブ ハブはそのすべてのポートでデバイスにメッセージを送信し、レイヤ 1 で動作しますが、アドレス テーブルを保持しません。

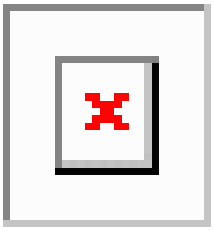
レイヤ 2 スイッチは、デバイス上のどのポートがそのポートだけに送信されたメッセージを受信するかを決定します。ただし、レイヤ 3 スイッチは、ARP キャッシュ（テーブル）を作成するデバイスです。

Reverse ARP

RFC 903 で定義された Reverse ARP（RARP）は、ARP と同じように動作しますが、RARP 要求パケットは MAC アドレスではなく IP アドレスを要求する点が異なります。RARP は多くの場合、ディスクレス ワークステーションで使用されます。これは、このタイプのデバイスには、起動時に使用する IP アドレスを格納する手段がないためです。認識できるアドレスは MAC アドレスだけで、これはハードウェアに焼き付けられているためです。

RARPを使用するには、ルータ インターフェイスとして、同じネットワーク セグメント上にRARPサーバが必要です。次の図に、RARP の仕組みを示します。

図 *5 : Reverse ARP*



RARPには、いくつかの制限があります。これらの制限により、ほとんどのビジネスでは、DHCP を使用してダイナミックに IP アドレスを割り当てています。DHCP は、RARP よりコスト効率が高く、必要な保守作業も少ないためです。最も重要な制限は次のとおりです。

- RARPはハードウェアアドレスを使用するため、多くの物理ネットワークを含む大規模なネットワークの場合は、各セグメント上に、冗長性のための追加サーバを備えた RARP サーバが必要です。各セグメントに 2 台のサーバを保持すると、コストがかかります。
- 各サーバは、ハードウェア アドレスと IP アドレスのスタティック マッピングのテーブルで設定する必要があります。IP アドレスの保守は困難です。
- RARP は、ホストの IP アドレスだけを提供し、サブネット マスクもデフォルト ゲートウェイも提供しません。

プロシキ ARP

プロキシ ARP を使用すると、物理的に 1 つのネットワーク上に存在するデバイスが、論理的に、同じデバイスまたはファイアウォールに接続された別の物理ネットワークの一部として表示されます。プロキシ ARP で、プライベート ネットワーク上のパブリック IP アドレスを持つデバイスをルータの背後に隠すと同時に、このデバイスを、ルータの前のパブリック ネットワーク上に表示できます。ルータはそのアイデンティティを隠すことにより、実際の宛先までパケットをルーティングする役割を担います。プロキシ ARP を使用すると、サブネット上のデバイスは、ルーティングもデフォルト ゲートウェイも設定せずにリモート サブネットまで到達できます。

複数のデバイスが同じデータリンク層のネットワークでなく、同じ IP ネットワーク内にある場合、これらのデバイスは相互に、ローカル ネットワーク上にあるかのようにデータを送信しようとします。ただし、これらのデバイスを隔てるルータは、ブロードキャスト メッセージを送信しません。これは、ルータがハードウェア レイヤのブロードキャストを渡さず、アドレスが解決されないためです。

デバイスでプロキシ ARP をイネーブルにし、ARP 要求を受信すると、プロキシ ARP はこれを、ローカル LAN 上にないシステムに対する要求と見なします。デバイスは、ブロードキャストの宛先であるリモートの宛先であるかのように、自身の MAC アドレスをリモートの宛先の IP アドレスに関連付ける ARP 応答で応答します。ローカルデバイスは、自身が宛先に直接、接続されていると認識していますが、実際には、そのパケットは、ローカルデバイスによりローカルサブネットワークから宛先のサブネットワークへと転送されています。デフォルトでは、プロキシ ARP はディセーブルになっています。

ローカル プロキシ ARP

ローカル プロキシ ARP を使用して、通常はルーティングが不要なサブネット内の IP アドレスを求める ARP 要求に対して、デバイスが応答できるようにすることができます。ローカルプロキシ ARP を有効にすると、ARP は、サブネット内の IP アドレスを求めるすべての ARP 要求に応答し、サブネット内のホスト間ですべてのトラフィックを転送します。この機能は、ホストが接続されているデバイスの設定により意図的に、ホストの直接通信が禁止されているサブネットだけで使用してください。

Gratuitous ARP

Gratuitous ARP は、送信元 IP アドレスと宛先 IP アドレスが同じである要求を送信し、重複する IP アドレスを検出します。Cisco NX-OSは Gratuitous ARP 要求または ARP キャッシュの更新の有効または無効をサポートします。

MAC 削除時の定期的な ARP 更新

ARP プロセスは MAC の削除を追跡し、設定されたカウントの設定された時間間隔で L3 VLAN インターフェイスに定期的な ARP 更新を送信します。MAC が学習されると、ARP プロセスは定期的な ARP 更新の送信を停止します。

詳細については、SVI の MAC 削除での定期的な ARP リフレッシュの構成（46 ページ） を参照してください。

収集スロットル

着信IPパケットがラインカードに転送されたときに、ネクストホップのアドレス解決プロトコル（ARP）の要求が解決されない場合、ラインカードはパケットをスーパーバイザに転送します（収集スロットル）。スーパーバイザはネクスト ホップの MAC アドレスを解決し、ハードウェアをプログラミングします。

ARP要求が送信されると、ソフトウェアは、同じネクストホップIPアドレスへのパケットがスーパーバイザに転送されないようにするために、ハードウェア内に /32 ドロップ隣接関係を追加します。ARP が解決されると、そのハードウェア エントリは正しい MAC アドレスで更新されます。タイムアウト期間が経過するまでにARPエントリが解決されない場合、そのエントリはハードウェアから削除されます。

（注） Glean スロットリングは IPv4 および IPv6 でサポートされますが、IPv6 リンクローカル アドレスはサポートされません。

パス MTU ディスカバリ

パス最大伝送ユニット（MTU）ディスカバリは、TCP接続のエンドポイント間のネットワーク内で使用可能な帯域幅の使用を最大化するための方法です。これはRFC 1191 で規定されています。この機能を有効または無効にしても、既存の接続に影響しません。

ICMP

Internet Control Message Protocol（ICMP）を使用して、IP 処理に関連するエラーおよびその他の情報を報告するメッセージ パケットを提供できます。ICMP は、ICMP 宛先到達不能メッセージ、ICMPエコー要求（2つのホスト間でパケットを往復送信する）、およびエコー返信メッセージなどのエラー メッセージを生成します。ICMP は多くの診断機能も備えており、ホストへのエラーパケットの送信およびリダイレクトが可能です。デフォルトでは、ICMPがイネーブルにされています。

次に示すのは、ICMP メッセージ タイプの一部です。

- ネットワーク エラー メッセージ
- ネットワーク輻輳メッセージ
- トラブルシューティング情報
- タイムアウト告知

（注）

- ICMP リダイレクトは、ローカル プロキシ ARP 機能がイネーブルになっているインターフェイスではディセーブルになります。
- ICMP リダイレクト機能は、Cisco N93-C64E-SG2-Q スイッチ プラットフォームではサポートされません。ICMP リダイレクトが Cisco N93-C64E-SG2-Q スイッチで現在有効になっている場合は、無効にする必要があります。

IPv4 の仮想化のサポート

IPv4 は、仮想ルーティング/転送（VRF）インスタンスをサポートします。

IPv4の前提条件

IPv4 には、次の前提条件があります。

- IPv4 はレイヤ 3 インターフェイス上だけで設定可能です。

IPv4 の注意事項および制約事項

IPv4 設定時の注意事項および制約事項は、次のとおりです。

- インターネット ピアリング モードに設定された Cisco Nexus 9300-EX および Cisco Nexus 9300-FX2 プラットフォーム スイッチには、完全な IPv4 および IPv6 インターネット ルートを同時にインストールするための十分なハードウェア容量がない場合があります。
- セカンダリ IP アドレスは、プライマリ IP アドレスの設定後にだけ設定できます。
- ローカル プロキシ ARP は、複数のサブネットに属する複数の HSRP グループを持つインターフェイスではサポートされません。
- **ip proxy-arp** コマンドは、 **fabric forwarding mode anycast-gateway**で有効になっている SVI の VXLAN EVPN ファブリックではサポートされていません。
- -R ライン カードを搭載した Cisco Nexus 9500-R プラットフォーム スイッチの場合、インターネット ピアリング モードは、グローバル インターネット ルーティング テーブルで配信されるプレフィックス パターンでのみ使用されます。このモードでは、他のプレフィックス配布/パターンは動作できますが、予測できません。その結果、プレフィックス パターンが実際のインターネット プレフィックス パターンである場合にのみ、達成可能な最大 LPM/LEM スケールが信頼できます。インターネット ピアリング モードでは、グローバル インターネット ルーティング テーブル内のルート プレフィックス パターン以外のルート プレフィックス パターンが使用されている場合、スイッチは文書化されたスケーラビリティの数値を正常に達成できない可能性があります。

- LPM の重いルーティングモードは、**9700**-EX、-FX、および-GX シリーズモジュールを搭載した Cisco Nexus **9500** シリーズスイッチでサポートされます。
- Cisco NX-OS リリース 10.2(3)F 以降、設定された間隔に基づいて IPv4 リダイレクト メッセージがトリガーされると、syslog が出力されます。
- Cisco NX-OS リリース 10.3(1)F 以降、スタティック ルーティングが Cisco Nexus 9808 スイッチでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、スタティック ルーティングは次のスイッチおよびライン カードでサポートされています。
  - Cisco Nexus 9804 スイッチ。
  - 9804 および 9808 プラットフォーム スイッチを搭載した Cisco Nexus X98900CD-A および X9836DM-A ライン カード。
- Cisco NX-OS リリース 10.3(1)F 以降、ダイナミック ルーティングが Cisco Nexus 9808 プラットフォーム スイッチでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、ダイナミック ルーティングは、9808 および 9804 スイッチを搭載した Cisco Nexus X98900CD-A および X9836DM-A ライン カードでサポートされます。
- Cisco NX-OS リリース 10.2(4)M 以降、MAC 削除サポートの定期的な ARP リフレッシュは、次の制限付きで Cisco Nexus 9000 シリーズ プラットフォーム スイッチで提供されます。
  - **ip arp refresh-adj-on-mac-delete retry** コマンドの構成中に、ARP が学習されて MAC が学習されていなくても、ARP プロセスはリフレッシュをトリガーしません。これは、MAC 削除/フラッシュ時に定期的な ARP リフレッシュを送信しようとします。
  - **ip arp refresh-adj-on-mac-delete retry**コマンドの構成後、MAC を削除すると、定期的な ARP リフレッシュ動作がトリガーされます。
  - この定期的な ARP リフレッシュのトリガーは、MAC 削除です。この機能は、バーストパケット受信時の MAC 学習ミスには対処しません。
  - 構成中に、規模/ネットワーク要件に基づいて適切な数と間隔を選択する必要があります。
- Cisco NX-OS リリース 10.4(1)F 以降、サブネット外の ARP 解決のサポートは、Cisco Nexus 9000 シリーズ プラットフォーム スイッチで次の L3 インターフェイスに提供されます。
  - イーサネット
  - サブインターフェイス
  - ポート チャネル
  - FEX
  - IP アンナンバード インターフェイス

（注）
- サブネット外 ARP 解決機能は、SVI L3 インターフェイス、および VPC、HSRP、または VXLAN 展開ではサポートされません。

- Cisco NX-OS リリース 10.4(2)F 以降では、次の機能を使用して、Cisco NX-OS デバイスのインターフェイスごとに ARP キャッシュ エントリを制限する **ip arp cache intf-limit** 構成がサポートされています。
  - グローバル モードとインターフェイス モードでサポートされます。ただし、インターフェイス モードの構成は、グローバル モードよりも優先されます。
  - 次の L3 インターフェイスでのみサポートされます。
    - SVI
    - SVI アンナンバード インターフェイス
  - 次の L3 インターフェイスではサポートされていません。
    - イーサネット
    - サブインターフェイス
    - ポート チャネル
    - アンナンバード インターフェイス
  - 構成がサポートされていないインターフェイスに適用される場合、この構成はグローバル モードに適用されます。
- Cisco NX-OS リリース 10.4(2)F 以降では、クラス E IP アドレスを使用してすべての NX-OS 機能を構成できます。IPv4 アドレス空間が不足している場合は、クラス E の IP アドレスを使用して機能を構成できます。クラス E IP アドレスでは、次のスイッチおよびライン カードがサポートされています。
  - Cisco Nexus 92348GC-X
  - Cisco Nexus 9300-EX/FX/FX2/FX3/H2R/H1/GX/GX2
  - Cisco Nexus 9300C
  - Cisco Nexus 9700-EX/FX/GX ライン カード

デフォルト設定

次の表に、IP パラメータのデフォルト設定値を示します。

パラメータ	デフォルト
ARP タイムアウト	1500 秒
『Proxy ARP』	無効化

IPv4 の設定

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

IPv4 アドレス指定の設定

ネットワーク インターフェイスにプライマリ IP アドレスを割り当てることができます。

手順の概要

1. **configure terminal**
2. **interface ethernet** *number*
3. **ip address** *ip-address/length* [*secondary*]
4. （任意） **show ip interface**
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**interface ethernet** *number*
	例：
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**ip address** *ip-address/length* [*secondary*] 例：	インターフェイスに対するプライマリ IPv4 アドレスまたはセカンダリ IPv4 アドレスを指定します。

	コマンドまたはアクション	目的
	`switch(config-if)# ip address 192.2.1.1 255.0.0.0`	• 4 分割ドット付き 10 進表記のアドレスでネットワーク マスクを指定します。たとえば、255.0.0.0 は、1 に等しい各ビットが、ネットワーク アドレスに属した対応するアドレス ビットを意味することを示します。 • ネットワーク マスクは、スラッシュ（/）および数字、つまり、プレフィックス長として示される場合もあります。プレフィックス長は、アドレスの高次の連続ビットのうち、何個がプレフィックス（アドレスのネットワーク部分）を構成しているかを指定する 10 進数値です。スラッシュは10進数値の前に置かれ、IPアドレスとスラッシュの間にスペースは入りません。
Step 4	（任意） **show ip interface** 例： `switch(config-if)# show ip interface`	IPv4 に設定されたインターフェイスを表示します。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

複数の IP アドレスの設定

セカンダリ IP アドレスは、プライマリ IP アドレスの設定後にのみ追加できます。

手順の概要

1. **configure terminal**
2. **interface ethernet** *number*
3. **ip address** *ip-address/length* [*secondary*]
4. （任意） **show ip interface**
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**interface ethernet** *number*
	例：
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**ip address** *ip-address/length* [*secondary*] 例： `switch(config-if)# ip address 192.168.1.1 255.0.0.0 secondary`	設定したアドレスをセカンダリ IPv4 アドレスとして指定します。
Step 4	（任意） **show ip interface** 例： `switch(config-if)# show ip interface`	IPv4用に設定されたインターフェイスを表示します。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

最大ホスト ルーティング モードの設定

デフォルトでは、Cisco NX-OS は階層方式で（モード 4 になるように設定されたファブリック モジュールとモード 3 になるように設定されたラインカード モジュールで）ルートをプログラミングし、デバイス上での最長プレフィクス照合（LPM）とホスト スケールが可能になります。

デフォルトの LPM およびホスト スケールを変更してシステム内のホストをさらにプログラミングできます。これは、ノードをレイヤ 2 ～ レイヤ 3 の境界ノードとして位置付けるときに必要になる場合があります。

（注） LPM テーブルのエントリをさらに拡大したい場合は、「非階層ルーティング モードの設定（Configuring Nonhierarchical Routing Mode (Cisco Nexus 9500 シリーズ スイッチのみ）」の項を参照して、ライン カード上のレイヤ 3 IPv4 および IPv6 ルートすべてをプログラミングしてファブリック モジュール上のルートはそのままにするようデバイスを設定します。

（注） この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

（注） 最大ホスト ルーティング モードのスケール数については、『Cisco Nexus 9000 Series NX-OS Verified Scalability Guide』を参照してください。

手順の概要

1. **configure terminal**
2. **[no] system routing max-mode host**
3. （任意） **show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**[no] system routing max-mode host** 例： `switch(config)# system routing max-mode host`	ライン カードを Broadcom T2 モード 2 に、ファブリック モジュールを Broadcom T2 モード 3 にして、サポートされるホスト数を増やします。
Step 3	（任意） **show forwarding route summary** 例： `switch(config)# show forwarding route summary`	LPM ルーティング モードを表示します。
Step 4	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。
Step 5	**reload** 例： `switch(config)# reload`	デバイス全体をリブートします。

非階層ルーティング モードの設定（Cisco Nexus 9500 プラットフォーム スイッチのみ）

ホストの規模が小さい場合（純粋なレイヤ 3 配置の場合など）、コンバージェンス パフォーマンスを向上させるために、ラインカードの最長プレフィクス照合（LPM）のルートをプログラミングすることを推奨します。そうすることによって、ラインカードのルートおよびホストがプログラミングされ、ファブリック モジュールのルートはプログラミングされません。

（注） この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

手順の概要

1. **configure terminal**
2. [**no**] **system routing non-hierarchical-routing** [**max-l3-mode**]
3. （任意） **show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	[**no**] **system routing non-hierarchical-routing** [**max-l3-mode**] 例： `switch(config)# system routing non-hierarchical-routing max-l3-mode`	ラインカードを Broadcom T2モード 3（または **max-l3-mode** オプションを使用している場合は Broadcom T2 モード 4）にし、より大きな LPM スケールをサポートします。その結果、IPv4 および IPv6 ルートのすべてが、ファブリック モジュールではなくラインカードでプログラミングされます。

Step 3	（任意） **show forwarding route summary**
	例：
	`switch(config)# show forwarding route summary`
	`Mode 3:`
	`120K IPv4 Host table`
	`16k LPM table (> 65 < 127 1k entry reserved)`
	`Mode 4:`
		LPM モードを表示します。

	コマンドまたはアクション	目的
	`16k V4 host/4k V6 host` `128k v4 LPM/20K V6 LPM`	
Step 4	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。
Step 5	**reload** 例： `switch(config)# reload`	デバイス全体をリブートします。

64 ビット ALPM ルーティング モードの設定（Cisco Nexus 9500 プラットフォーム スイッチのみ）

64 ビットアルゴリズム最長プレフィックス一致（ALPM）機能を使用して、IPv4 および IPv6 ルートテーブルエントリを管理できます。64 ビット ALPM ルーティング モードでは、デバイスに保存できるルート エントリの数が増加します。このモードでは、次のいずれかをプログラムできます。

- 80,000 IPv6 エントリ、IPv4 エントリなし
- IPv6 エントリなし、128,000 の IPv4 エントリ
- x 個の IPv6 エントリと IPv4 エントリ（$2x + y$ の場合）

（注） この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

（注） 64 ビット ALPM ルーティング モードのスケール数については、『Cisco Nexus 9000 シリーズ NX-OS 検証済みスケーラビリティ ガイド』を参照してください。

手順の概要

1. **configure terminal**
2. **[no] system routing mode hierarchical 64b-alpm**
3. （任意） **show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**[no] system routing mode hierarchical 64b-alpm** 例： `switch(config)# system routing mode hierarchical 64b-alpm`	マスク長が 64 以下のすべての IPv4 および IPv6 LPM ルートをファブリックモジュールにプログラミングします。IPv4 および IPv6 のすべてのホスト ルート、およびマスク長が 65 ～ 127 であるすべての LPM ルートがライン カードでプログラミングされます。
Step 3	（任意） **show forwarding route summary** 例： `switch(config)# show forwarding route summary`	LPM モードを表示します。
Step 4	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。
Step 5	**reload** 例： `switch(config)# reload`	デバイス全体をリブートします。

ALPM ルーティング モードの設定（Cisco Nexus 9300 プラットフォーム スイッチのみ）

Cisco Nexus 9300 プラットフォーム スイッチは、多数の LPM ルート エントリをサポートするように設定できます。

（注） この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

（注） ALPM ルーティング モードのスケール数については、『Cisco Nexus 9000 シリーズ NX-OS 検証済みスケーラビリティ ガイド』を参照してください。

手順の概要

1. **configure terminal**
2. **[no] system routing max-mode l3**
3. （任意） **show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**[no] system routing max-mode l3** 例： `switch(config)# system routing max-mode l3`	デバイスを Broadcom T2 モード 4にして、より大きな LPM スケールをサポートします。
Step 3	（任意） **show forwarding route summary** 例： `switch(config)# show forwarding route summary`	LPM モードを表示します。
Step 4	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。
Step 5	**reload** 例： `switch(config)# reload`	デバイス全体をリブートします。

LPMヘビールーティングモードの設定（CiscoNexus9200および9300-EXプラットフォーム スイッチおよび 9732C-EX ライン カードのみ）

Cisco NX-OS リリース 7.0(3)I4(4) 以降では、より多くの LPM ルート エントリをサポートするために LPM のヘビー ルーティング モードを設定できます。このルーティング モードをサポートするのは、Cisco Nexus 9200 および 9300-EX プラットフォーム スイッチと、9732C-EX ライン カードを搭載した Cisco Nexus 9508 スイッチだけです。

（注） この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

（注） LPM ヘビー ルーティング モードのスケール数については、『Cisco Nexus 9000 Series NX-OS Verified Scalability Guide』を参照してください。

手順の概要

1. **configure terminal**
2. **[no] system routing template-lpm-heavy**
3. （任意） **show system routing mode**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**[no] system routing template-lpm-heavy** 例： `switch(config)# system routing template-lpm-heavy`	デバイスを LPM ヘビー ルーティング モードにして、より大きな LPM スケールをサポートします。

Step 3	（任意） **show system routing mode**
	例：
	`switch(config)# show system routing mode`
	`Configured System Routing Mode: LPM Heavy`
	`Applied System Routing Mode: LPM Heavy`
		LPM ルーティング モードを表示します。

Step 4	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。
Step 5	**reload** 例： `switch(config)# reload`	デバイス全体をリブートします。

LPM インターネット ピアリング ルーティング モードの設定

Cisco NX-OS リリース7.0(3)I6(1) 以降では、IPv4 および IPv6 LPM インターネット ルート エントリをサポートするために LPM インターネット ピアリング ルーティング モードを設定できます。このモードは、IPv4 プレフィックス（/32 までのプレフィックス長）および IPv6 プレフィックス（/83 までのプレフィックス長）のダイナミック トライ（ツリー ビット ルックアップ）をサポートします。

Cisco NX-OS リリース 9.3(1) 以降、Cisco Nexus 9500-R プラットフォーム スイッチはこのルーティング モードをサポートします。

（注） この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

（注） LPM インターネット ピアリング ルーティング モードのスケール数については、『Cisco Nexus 9000 Series NX-OS Verified Scalability Guide』を参照してください。LPM インターネット ピアリング モードの Cisco Nexus 9500-R プラットフォーム スイッチは、インターネット ピアリング プレフィックスを使用する場合にのみ、予測どおりにスケールアウトします。Cisco Nexus 9500-R プラットフォーム スイッチが他のプレフィックス パターンを使用している場合は、文書化されたスケーラビリティの数値を達成できない可能性があります。

手順の概要

1. **configure terminal**
2. [**no**] **system routing template-internet-peering**
3. （任意） **show system routing mode**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

	コマンドまたはアクション	目的
Step 2	[**no**] **system routing template-internet-peering** 例： `switch(config)# system routing template-internet-peering`	デバイスを LPM インターネット ピア ルーティング モードにして、IPv4 および IPv6 LPMインターネット ルート エントリをサポートします。

Step 3	（任意） **show system routing mode**
	例：
	`switch(config)# show system routing mode`
	`Configured System Routing Mode: Internet Peering`
	`Applied System Routing Mode: Internet Peering`
		LPM ルーティング モードを表示します。

Step 4	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。
Step 5	**reload** 例： `switch(config)# reload`	デバイス全体をリブートします。

LPM デュアルホスト ルーティング モードの構成

Cisco NX-OS リリース 7.0(3)I5(1) 以降では、ARP/ND スケールをデフォルト モード値の 2 倍に増やすために LPM デュアル ホスト ルーティング モードを設定できます。このルーティング モードをサポートするのは、Cisco Nexus 9200 および 9300-EX プラットフォーム スイッチだけです。

Cisco NX-OS リリース 10.3(1)F 以降、**system routing template-dual-stack-host-scale** プロファイルは、Cisco Nexus 9300-FX3/GX/GX2B ToR スイッチおよび Nexus 9408 スイッチでマルチキャストと VXLAN をサポートします。

（注） **system routing template-dual-stack-host-scale** プロファイルが BGW で使用されていないことを確認します。

（注） この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

（注） LPM ルーティング モードのスケール数については、『』『**Cisco Nexus 9000 シリーズ NX-OS 検証済みスケーラビリティ ガイド**』を参照してください。

手順の概要

1. **configure terminal**
2. [**no**] **system routing template-dual-stack-host-scale**
3. （任意） **show system routing mode**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	[**no**] **system routing template-dual-stack-host-scale**
	例：
	`switch(config)# system routing template-dual-stack-host-scale`
	`Warning: The command will take effect after next reload.`
	`Note: This requires copy running-config to startup-config before switch reload.`
		デバイスを LPM デュアルホスト ルーティング モードにして、より大きな ARP/ND スケールをサポートします。

Step 3	（任意） **show system routing mode** 例： `switch(config)# show system routing mode`	LPM ルーティング モードを表示します。
Step 4	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。
Step 5	**reload** 例： `switch(config)# reload`	デバイス全体をリブートします。

スタティック ARP エントリの設定

デバイス上でスタティック ARP エントリを設定して、IP アドレスをスタティック マルチキャスト MAC アドレスを含む MAC ハードウェア アドレスにマッピングできます。

手順の概要

1. **configure terminal**
2. **interface ethernet** *number*
3. **ip arp address** *ip-address mac-address*
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**interface ethernet** *number*
	例：
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**ip arp address** *ip-address mac-address* 例： `switch(config-if)# ip arp 192.168.1.1 0019.076c.1a78`	IP アドレスを MAC アドレスにスタティック エントリとして関連付けます。
Step 4	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

プロキシ ARP の設定

デバイス上でプロキシ ARP を設定して、他のネットワークまたはサブネット上のホストのメディア アドレスを決定します。

手順の概要

1. **configure terminal**
2. **interface ethernet** *number*
3. **ip proxy arp**
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**interface ethernet** *number*
	例：
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**ip proxy arp** 例： `switch(config-if)# ip proxy arp`	インターフェイス上でプロキシ ARP を有効にします。
Step 4	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

イーサネット インターフェイスでのローカル プロキシ ARP の設定

イーサネット インターフェイス上でローカル プロキシ ARP を設定することができます。

手順の概要

1. **configure terminal**
2. **interface ethernet** *number*
3. [**no**]**ip local-proxy-arp**
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的

Step 2	**interface ethernet** *number*
	例：
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**[no]ip local-proxy-arp** 例： `switch(config-if)# ip local-proxy-arp`	インターフェイス上でローカル プロキシ ARP をイネーブルにします。
Step 4	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

SVI でのローカル プロキシ ARP の設定

SVI でローカル プロキシ ARP を設定できます。CiscoNX-OS リリース 7.0(3)I7(1) 以降では、対応する VLAN で ARP ブロードキャストを抑制することができます。

始める前に

ARP ブロードキャストを抑制する場合は、hardware access-list tcam region arp-ether 256 double-wide コマンドを使用して、ARP/レイヤ 2 Ethertype の倍幅 ACL TCAM リージョン サイズを設定し、設定を保存して、スイッチをリロードします。（詳細については『Cisco Nexus 9000 シリーズ NX-OS セキュリティ設定ガイド』の「ACL TCAM リージョンサイズの設定」のセクションを参照してください。）

手順の概要

1. **configure terminal**
2. **interface vlan vlan-id**
3. **[no] ip local-proxy-arp [no-hw-flooding]**
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的

Step 2	**interface vlan vlan-id**
	例：
	`switch(config)# interface vlan 5`
	`switch(config-if)#`
		VLAN インターフェイスを作成し、SVI の設定モードを開始します。

Step 3	[**no**] **ip local-proxy-arp** [**no-hw-flooding**] 例： `switch(config-if)# ip local-proxy-arp no-hw-flooding`	SVI でローカル プロキシ ARP をイネーブルにします。no-hw-flooding オプションは、対応する VLAN での ARP ブロードキャストを抑制します。 （注） no-hw-flooding オプションを設定し、SVI で ARP ブロードキャストを許可するように設定を変更する場合は、まず no ip local-proxy-arp no-hw-flooding コマンドを使用してこの機能を無効にして、ip local-proxy-arp コマンドを開始する必要があります。
Step 4	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

SVI の MAC 削除での定期的な ARP リフレッシュの構成

Cisco NX-OS リリース 10.2(4)M 以降、SVI の MAC 削除時に定期的な ARP リフレッシュを行うよう構成できます。

デフォルトでは、このコマンドは無効になっています。このコマンドは、定期的な ARP リフレッシュの SVI で設定して、MAC 削除/フラッシュでサイレントホストの ARP 応答パケットから MAC を学習する必要があります。

手順の概要

1. **configure terminal**
2. **interface vlan vlan-id**
3. [**no**] **ip arp refresh-adj-on-mac-delete retry** [**count** <*frequency count*>] [**interval** <*interval in sec*>]
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**interface vlan vlan-id**
	例：
	`switch(config)# interface vlan 5`
	`switch(config-if)#`
		VLAN インターフェイスを作成し、SVI の設定モードを開始します。

Step 3	[**no**] **ip arp refresh-adj-on-mac-delete retry** [**count** <*frequency count*>] [**interval** <*interval in sec*>]
	例：
	`switch(config-if)# ip arp refresh-adj-on-mac-delete retry count 3 interval 15`
	`switch(config-if)#`
		MAC 削除/フラッシュでサイレントホストの ARP 応答パケットから MAC を学習するように ARP リフレッシュを構成します。
		• <*frequency count*>：範囲は 1 ～ 3 です。デフォルトは 3 です。
		• <*interval in sec*>：範囲は 1 ～ 60 秒です。デフォルトは 15 秒です。
		（注）
		間隔が ARP リフレッシュ時間の 3/4 より大きい場合、このコマンドは拒否され、次のメッセージが表示されます：
		`ARP タイムアウト構成により、ARP リフレッシュはこの間隔よりも早く送信されます。この構成は役に立ちません。`

Step 4	（任意） **copy running-config startup-config**
	例：
	`switch(config-if)# copy running-config startup-config`
	`switch(config-if)#`
		実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

無償 ARP の設定

インターフェイス上で Gratuitous ARP を設定できます。

手順の概要

1. **configure terminal**

2. **interface ethernet** *number*
3. **ip arp gratuitous** {**request** | **update**]
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**interface ethernet** *number*
	例：
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**ip arp gratuitous** {**request** | **update**] 例： `switch(config-if)# ip arp gratuitous request`	インターフェイス上で無償 ARP をイネーブルにします。無償 ARP はデフォルトで有効になっています。
Step 4	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

サブネット外の ARP 解決の構成

Cisco NX-OS リリース 10.4(1)F 以降では、**ip arp outside-subnet** コマンドを使用してサブネット外 ARP 解決を有効または無効にできます。

このコマンドは、グローバルモードとインターフェイスモードの両方で使用できます。このコマンドが有効になっている場合、config-replace およびデュアル ステージ コミットには影響しません。

（注） このコマンドを有効にすると、Cisco NX-OS リリース 10.4(1)F からのダウングレードが制限され、ダウングレードを続行する前に、サブネット外 ARP 解決構成を削除するように求めるエラーメッセージがユーザーに表示されます。

手順の概要

1. **configure terminal**
2. **[no] ip arp outside-subnet**
3. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**[no] ip arp outside-subnet** 例： `switch(config)# ip arp outside-subnet`	接続されたホストのサブネット パケット トランザクションからの ARP を有効または無効にします。
Step 3	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

SVI インターフェイスごとの ARP キャッシュの構成

Cisco NX-OS リリース 10.4(2)F 以降では、Cisco NX-OS デバイスの SVI インターフェイスごとに許可される ARP キャッシュ エントリの最大数を設定できます。この構成は、グローバル モードとインターフェイス モードの両方でサポートされます。

手順の概要

1. **configure terminal**
2. **interface vlan** *vlan-id*
3. **[no] ip arp cache intf-limit** *value*
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**interface vlan** *vlan-id*
	例：
	`switch(config)# interface vlan 5`
	`switch(config-if)#`
		VLAN インターフェイスを作成し、SVI の設定モードを開始します。

Step 3	**[no] ip arp cache intf-limit** *value*
	例：
	`switch(config-if)# ip arp cache intf-limit 50000`
	`switch(config-if)#`
		SVI インターフェイスの ARP キャッシュ エントリの制限を構成します。有効な ARP エントリの範囲は 1 ～ 128000 です。
		intf-limit：インターフェイスごとの有効なダイナミック ARP エントリの数を指定します。
		構成を削除するには、この **no** コマンドの no 形式を使用します。

Step 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

パス MTU ディスカバリの設定

パス MTU ディスカバリを設定できます。

手順の概要

1. **configure terminal**
2. **ip tcp path-mtu-discovery**
3. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**ip tcp path-mtu-discovery** 例： `switch(config)# ip tcp path-mtu-discovery`	パス MTU ディスカバリをイネーブルにします。
Step 3	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

IP ダイレクト ブロードキャストの設定

IP ダイレクト ブロードキャストは、宛先アドレスが何らかの IP サブネットの有効なブロードキャスト アドレスであるにもかかわらず、その宛先サブネットに含まれないノードから発信される IP パケットです。

宛先サブネットに直接接続されていないデバイスは、そのサブネット上のホストを宛先とするユニキャスト IP パケットを転送する場合と同じ方法で IP ダイレクト ブロードキャストを転送します。ダイレクト ブロードキャスト パケットが、宛先サブネットに直接接続されたデバイスに到着すると、そのパケットはその宛先サブネット上でブロードキャストされます。パケットの IP ヘッダー内の宛先アドレスはそのサブネットに設定された IP ブロードキャスト アドレスに書き換えられ、パケットはリンク層ブロードキャストとして送信されます。

あるインターフェイスでダイレクト ブロードキャストがイネーブルになっている場合、着信した IP パケットが、そのアドレスに基づいて、そのインターフェイスが接続されているサブネットを対象とするダイレクト ブロードキャストとして識別されると、そのパケットはそのサブネット上でブロードキャストされます。アクセス リストを通じて渡すこれらパケットのみがサブネット上でブロードキャストされるように、IP アクセス リストを通じてこれらブロードキャストを任意でフィルタリングすることができます。

IP ダイレクト ブロードキャストをイネーブルにするには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **ip directed-broadcast** [*acl*]

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**ip directed-broadcast** [*acl*] 例： `switch(config-if) # ip directed-broadcast`	ダイレクト ブロードキャストの物理ブロードキャストへの変換をイネーブルにします。IP アクセス リスト上のこれらのブロードキャストを任意でフィルタリングできます。

IP 収集スロットルの設定

IP 収集スロットルを設定して、到達できないかまたは存在しないネクスト ホップの ARP 解決のためにスーパーバイザに送信される不要な収集パケットをフィルタリングすることを推奨します。IP 収集スロットルは、ソフトウェアのパフォーマンスを向上させ、トラフィックをより効率的に管理します。

（注） Glean スロットリングは IPv4 および IPv6 でサポートされますが、IPv6 リンクローカル アドレスはサポートされません。

手順の概要

1. **configure terminal**
2. [**no**] **hardware ip glean throttle**
3. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	[**no**] **hardware ip glean throttle** 例： `switch(config) # hardware ip glean throttle`	IP 収集スロットルをイネーブルにします。

	コマンドまたはアクション	目的
Step 3	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

ハードウェア IP 収集スロットルの最大値の設定

転送情報ベース（FIB）にインストールされている隣接関係の最大ドロップ数を制限できます。

手順の概要

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

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	[**no**] **hardware ip glean throttle maximum** *count* 例： `switch(config) # hardware ip glean throttle maximum 2134`	FIB にインストールされるドロップ隣接関係の数を設定します。
Step 3	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

ハードウェア IP 収集スロットルのタイムアウトの設定

インストールされたドロップ隣接関係が FIB 内に残る時間のタイムアウトを設定できます。

手順の概要

1. **configure terminal**

2. [**no**] **hardware ip glean throttle maximum timeout** *timeout-in-seconds*
3. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	[**no**] **hardware ip glean throttle maximum timeout** *timeout-in-seconds* 例： `switch(config)# hardware ip glean throttle maximum timeout 300`	インストールされたドロップ隣接関係が FIB 内に残る時間のタイムアウトを設定します。 範囲は 300 秒（5 分）～ 1800 秒（30 分）です。 （注） タイムアウト期間を超えた後、ドロップ隣接関係は FIB から削除されます。
Step 3	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

ICMP 送信元 IP フィールドのインターフェイス IP アドレスの設定

ICMP エラー メッセージを処理するように ICMP ソース IP フィールドのインターフェイス IP アドレスを設定できます。

手順の概要

1. **configure terminal**
2. [**no**] **ip source** {**ethernet** *slot/port* | **loopback** *number* | **port-channel** *number*} **icmp-errors**
3. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	[**no**] **ip source** {**ethernet** *slot/port* | **loopback** *number* | **port-channel** *number*} **icmp-errors** 例： `switch(config)# ip source loopback 0 icmp-errors`	ICMP 送信元 IP フィールドのインターフェイス IP アドレスを設定し、ICMPエラーメッセージをルーティングします。
Step 3	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

IPv4 リダイレクト Syslog の構成

IPv4 リダイレクト Syslog を有効/無効にするか、ログ間隔を変更するには、次の CLI を使用します。

（注） デフォルトでは、syslog のリダイレクトが有効になっています。

手順の概要

1. **configure terminal**
2. **ip redirect syslog** [<*value*>]
3. （任意） **no ip redirect syslog**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例：	グローバルコンフィギュレーションモードを開始します

	コマンドまたはアクション	目的
	`switch# configure terminal` `switch(config)#`	

Step 2	**ip redirect syslog** [<*value*>]
	例：
	`switch(config)# ip redirect syslog 60`
	`switch(config)#`
		過剰な IP リダイレクトメッセージの syslog を設定します。
		• **ip redirect syslog**: IPv4 リダイレクト メッセージの syslog を有効にします。
		• *value*: ログ間隔を設定します。範囲は最小 30 秒から最大 1800 秒です。デフォルトインターバルは 60 秒です。

Step 3	（任意） **no ip redirect syslog** 例： `switch(config)# no ip redirect syslog`	過剰な IPv4 リダイレクトメッセージの syslog を無効にします。

IPv4 設定の確認

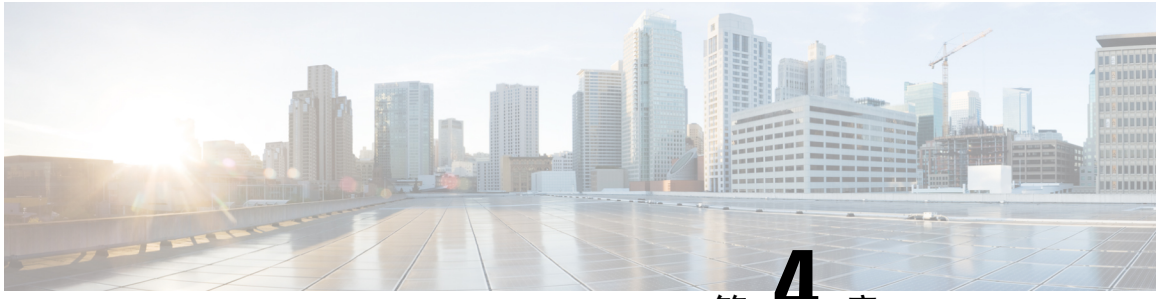
IPv4 の設定情報を表示するには、次のいずれかの作業を行います。

コマンド	目的
show ip adjacency	隣接関係テーブルを表示します。
show ip adjacency summary	スロットル隣接関係の数のサマリーを表示します。
show ip arp	ARP テーブルを表示します。
show ip arp summary	スロットル隣接関係の数のサマリーを表示します。
show ip interface	IP に関連するインターフェイス情報を表示します。
show ip arp statistics [**vrf** *vrf-name*]	ARP 統計情報を表示します。
show ip arp internal info interface <*interface-name*>	設定されたカウントと間隔を表示します

その他の参考資料

IPv4 の関連資料

関連項目	マニュアル タイトル
TCAM リージョン	詳細については『Cisco Nexus 9000 シリーズセキュリティ設定ガイド』の「ACL TCAM リージョン サイズの設定」のセクションを参照してください。



第 4 章

IPv6 の設定

この章は次のトピックで構成されています。

- IPv6 について （59 ページ）
- 仮想化のサポート （80 ページ）
- ECMP を使用した IPv6 ルート （80 ページ）
- IPv6の前提条件 （80 ページ）
- IPv6 の注意事項および制約事項 （80 ページ）
- IPv6 の設定 （82 ページ）
- IPv6 設定の確認 （103 ページ）
- IPv6 の設定例 （103 ページ）

IPv6 について

IPv6 は、IPv4 の後継として設計されており、ネットワーク アドレス ビット数が 32 ビット（IPv4 の場合）から 128 ビットに増やされています。IPv6 は IPv4 に基づいていますが、アドレス空間が大幅に拡大されており、メイン ヘッダーと拡張ヘッダーの簡素化など、その他の機能強化が含まれています。

拡大された IPv6 アドレス空間により、ネットワークのスケーラビリティが可能となり、グローバルな到達可能性が提供されます。簡素化された IPv6 パケット ヘッダー形式により、パケットの処理効率が向上しています。柔軟性の高い IPv6 アドレス空間により、プライベート アドレスの必要性と、プライベート（グローバルに一意ではない）アドレスを限られた数のパブリック アドレスに変換するネットワーク アドレス変換（NAT）の使用が削減されます。IPv6 を使用すると、ネットワークの境界にある境界ルータによる特別な処理を必要としない新しいアプリケーション プロトコルがイネーブルになります。

プレフィックス集約、簡易ネットワーク再番号割り当て、IPv6 サイト マルチホーミング機能などの IPv6 機能により、さらに効率的にルーティングが行われます。IPv6 は、Routing Information Protocol（RIP）、Integrated Intermediate System-to-Intermediate System（IS-IS）、IPv6 向け Open Shortest Path First（OSPF）、マルチプロトコル Border Gateway Protocol（BGP）をサポートしています。

IPv6 アドレス形式

IPv6 アドレスは 128 ビットつまり 16 バイトです。このアドレスは、x:x:x:x:x:x:x:x のように、コロン（:）で区切られた 16 ビット 16 進数のブロック 8 つに分かれています。

次に、IPv6 アドレスの例を 2 つ示します。

```
2001:0DB8:7654:3210:FEDC:BA98:7654:3210
```

```
2001:0DB8:0:0:8:800:200C:417A
```

IPv6 アドレスの中には、連続するゼロが含まれます。IPv6 アドレスの先頭、中間、または末尾で、この連続するゼロの代わりに2つのコロン（::）を使用できます。次の表は、圧縮された IPv6 アドレス フォーマットの一覧です。

（注） IPv6 アドレスでは、アドレス中で最も長く連続するゼロの代わりに、2 つのコロン（::）を 1 度だけ使用できます。

連続する 16 ビット値がゼロで示されている場合は、2 つのコロンを IPv6 アドレスの一部として使用できます。インターフェイスごとに複数の IPv6 アドレスを設定できますが、設定できるリンクローカル アドレスは 1 つだけです。

IPv6 アドレス中の 16 進数の文字の大文字と小文字は区別されません。

表 *7*: 圧縮された *IPv6* アドレス形式

IPv6 アドレス タイプ	優先形式	圧縮形式
ユニキャスト	2001:0:0:0:0:DB8:800:200C:417A	2001::0DB8:800:200C:417A
マルチキャスト	FF01:0:0:0:0:0:0:101	FF01::101
ループバック	0:0:0:0:0:0:0:1	::1
未指定	0:0:0:0:0:0:0:0	::

ノードは表にあるループバック アドレスを使用して、IPv6 パケットを自分宛てテーブルにに送信できます。IPv6 のループバック アドレスは、IPv4 のループバック アドレスと同じです。詳細については、概要 （3 ページ）を参照してください。

（注） IPv6 ループバック アドレスは、物理インターフェイスに割り当てることはできません。送信元または宛先のアドレスとして IPv6 ループバック アドレスを含むパケットは、そのパケットを作成したノードの外には転送できません。IPv6 ルータは、IPv6 ループバック アドレスを送信元アドレスまたは宛先アドレスとするパケットを転送しません。

（注） IPv6 未指定アドレスは、インターフェイスに割り当てることはできません。未指定 IPv6 アドレスは、IPv6 パケット内の宛先アドレスまたは IPv6 ルーティング ヘッダーとして使用しないでください。

IPv6 プレフィックスは、RFC 2373 で規定された形式です。この形式では、IPv6 アドレスが、コロンに囲まれた 16 ビット値を使用した 16 進数で指定されています。プレフィックス長は、アドレスの高次の連続ビットのうち、何個がプレフィックス（アドレスのネットワーク部分）を構成しているかを指定する 10 進数値です。たとえば、2001:0DB8:8086:6502::/32 は有効な IPv6 プレフィックスです。

IPv6 ユニキャスト アドレス

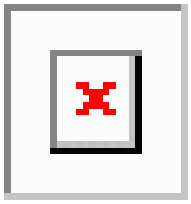
IPv6 ユニキャスト アドレスは、1 つのノード上の 1 つのインターフェイスの ID です。ユニキャスト アドレスに送信されたパケットは、そのアドレスが示すインターフェイスに配信されます。

集約可能グローバル アドレス

集約可能グローバルアドレスは、集約可能なグローバルユニキャストプレフィックスによる IPv6 アドレスです。集約可能グローバル ユニキャスト アドレスの構造により、グローバル ルーティング テーブル内のルーティング テーブル エントリ数を制限するルーティング プレフィックスの厳密な集約が可能になります。集約可能グローバル アドレスは、組織を上に向かって、最終的にインターネット サービス プロバイダー（ISP）まで集約されるリンク上で使用されます。

集約可能なグローバル IPv6 アドレスは、グローバル ルーティング プレフィックス、サブネット ID、およびインターフェイス ID により定義されます。バイナリ 000 で始まるアドレスを除き、グローバルユニキャストアドレスはすべて 64 ビットインターフェイス ID を持ちます。IPv6 グローバルユニキャストアドレスの割り当てには、バイナリ値 001（2000::/3）から始まるアドレスの範囲が使用されます。次の図は、集約可能グローバル アドレスの構造を示しています。

図 *6*：集約可能グローバル アドレス形式



2000::/3（001）～ E000::/3（111）のプレフィックスを持つアドレスには、Extended Universal Identifier（EUI）64 形式の 64 ビットインターフェイス識別子が必要です。インターネット割り当て番号局（IANA）は、2000::/16 の範囲の IPv6 アドレス空間を地域レジストリに割り当てます。

集約可能なグローバル アドレスは、48 ビット グローバル ルーティング プレフィックスと、16 ビット サブネット ID または Site-Level Aggregator（SLA）で構成されます。IPv6 集約可能グローバルユニキャストアドレスの形式に関するドキュメント（RFC 2374）によると、グローバル ルーティング プレフィックスには、Top-Level Aggregator（TLA）と Next-Level Aggregator（NLA）と

いう 2 つの階層構造のフィールドが含まれています。TLS フィールドおよび NLA フィールドはポリシーベースであるため、IETF は、これらのフィールドを RFC から削除することを決定しました。この変更以前に展開された既存の IPv6 ネットワークの中には、依然として、古いアーキテクチャ上のネットワークを使用しているものもあります。

個々の組織は、16 ビット サブネット フィールドであるサブネット ID を使用して、ローカル アドレス指定階層を作成したり、サブネットを識別したりできます。サブネット ID は IPv4 でのサブネットに似ていますが、IPv6 サブネット ID を持つ組織では最大 65,535 個のサブネットをサポートできるという点が異なります。

インターフェイス ID により、リンク上のインターフェイスが識別されます。インターフェイス ID は、リンク上では一意です。多くの場合、インターフェイス ID は、インターフェイスのリンク層アドレスと同じか、リンク層アドレスに基づいています。集約可能なグローバル ユニキャストやその他の IPv6 アドレス タイプで使用されるインターフェイス ID は 64 ビットであり、形式は変更済み EUI-64 フォーマットです。

インターフェイス ID は、次のいずれかに該当する修正 EUI-64 形式です。

- すべての IEEE 802 インターフェイス タイプ（イーサネット、およびファイバ分散データ インターフェイスなど）の場合は、最初の 3 オクテット（24 ビット）がそのインターフェイスの 48 ビット リンク層アドレス（MAC アドレス）の Organizationally Unique Identifier（OUI）、4 番めと 5 番めのオクテット（16 ビット）が FFFE の固定 16 進数値、そして、最後の 3 オクテット（24 ビット）が MAC アドレスの最後の 3 オクテットです。最初のオクテットの 7 番めのビットである Universal/Local（U/L）ビットの値は 0 または 1 です。ゼロはローカルに管理されている ID を表し、1 はグローバルに一意の IPv6 インターフェイス ID を表します。
- その他のすべてのインターフェイス タイプ（シリアル、ループバック、ATM、フレーム リレー種別など）の場合、インターフェイス ID は IEEE 802 インターフェイス タイプのインターフェイス ID に似ていますが、ルータの MAC アドレス プールからの最初の MAC アドレスが ID として使用される点が異なります（インターフェイスが MAC アドレスを持たないため）。

（注） PPP（ポイントツーポイント プロトコル）を使用するインターフェイスの場合は、接続の両端のインターフェイスが同じ MAC アドレスを持つため、接続の両端のインターフェイス ID が、両方の ID が一意となるまでネゴシエートされます（ランダムに選択され、必要に応じて再構築されます）。ルータの最初の MAC アドレスが、PPP を使用するインターフェイスの ID として使用されます。

ルータに IEEE 802 インターフェイス タイプがない場合は、ルータのインターフェイスでリンク ローカル IPv6 アドレスが次のシーケンスで生成されます。

1. ルータに MAC アドレスが（ルータの MAC アドレス プールから）照会されます。
2. 使用可能な MAC アドレスがルータにない場合は、ルータのシリアル番号を使用してリンク ローカル アドレスが作成されます。

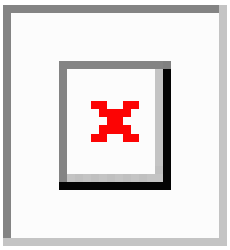
3. リンクローカル アドレスの作成にルータのシリアル番号を使用できない場合、ルータは MD5 ハッシュを使用して、ルータのホスト名からルータの MAC アドレスを決定します。

リンクローカル アドレス

リンクローカル アドレスは、リンクローカル プレフィックス FE80::/10（1111 1110 10）と変更された EUI-64 形式のインターフェイス識別子を使用するすべてのインターフェイスを自動的に設定できる IPv6 ユニキャスト アドレスです。ネイバー探索プロトコル（NDP）およびステートレス自動設定プロセスでは、リンクローカルアドレスが使用されます。ローカルリンク上のノードは、リンクローカル アドレスを使用して通信できます。ノードの通信にグローバルに一意のアドレスは不要です。次の図は、以下のリンクローカル アドレスの構造を示しています。

IPv6 ルータは、送信元または宛先がリンクローカル アドレスであるパケットを他のリンクに転送できません。

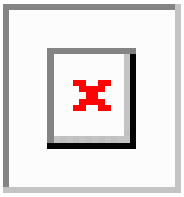
図 *7:* リンクローカル アドレス形式



IPv4 互換 IPv6 アドレス

IPv4 互換 IPv6 アドレスは、アドレスの上位 96 ビットがゼロであり、アドレスの下位 32 ビットが IPv4 アドレスである IPv6 ユニキャスト アドレスです。IPv4 互換 IPv6 アドレスの形式は、0:0:0:0:0:0:A.B.C.D または ::A.B.C.D です。IPv4 互換 IPv6 アドレスの 128 ビット全体がノードの IPv6 アドレスとして使用され、下位 32 ビットに埋め込まれた IPv4 アドレスがノードの IPv4 アドレスとして使用されます。IPv4 互換 IPv6 アドレスは、IPv4 と IPv6 の両方のプロトコル スタックをサポートするノードに割り当てられ、自動トンネルで使用されます。図に、IPv4 互換 IPv6 アドレスの構造と、許容されるいくつかのアドレス形式を示します。

図 *8 : IPv4* 互換 *IPv6* アドレス形式



ユニーク ローカル アドレス

一意のローカルアドレスは、グローバルに一意であり、ローカル通信を目的とした IPv6 ユニキャスト アドレスです。グローバルなインターネット上でのルーティングには対応しておらず、サイトなどの限られたエリア内だけでルーティング可能です。限られた複数のサイト間もルーティン

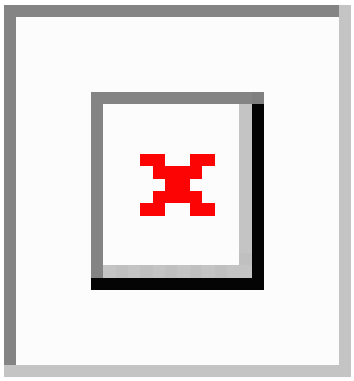
グできる場合もあります。アプリケーションは、一意のローカルアドレスをグローバルスコープのアドレスのように扱うことができます。

一意のローカル アドレスには、次の特性があります。

- グローバルに一意のプレフィックスを持っている（一意である可能性が大）。
- 既知のプレフィックスがあるため、サイト境界で簡単にフィルタリングできる。
- アドレス競合を発生させたり、これらのプレフィックスを使用するインターフェイスのリナンバリングを必要としたりすることなく、サイトを結合またはプライベートに相互接続できる。
- ISP に依存せず、永続的または断続的なインターネット接続がなくてもサイト内での通信に使用できる。
- ルーティングやドメイン ネーム サーバ（DNS）を通して誤ってサイト外に漏れても、他のどのアドレスとも競合しない。

図に、一意のローカル アドレスの構造を示します。

図 *9*: ユニーク ローカル アドレスの構造



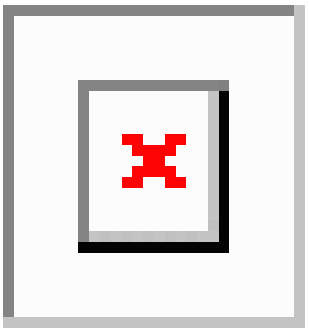
サイト ローカル アドレス

RFC 3879 によりサイトローカル アドレスの使用が廃止されたため、プライベート IPv6 アドレスの設定時には、RFC 4193 で推奨されるユニーク ローカル アドレス（UCA）を使用する必要があります。

IPv4 パケット ヘッダー

基本 IPv4 パケット ヘッダーには、合計サイズが 20 オクテット（160 ビット）の 12 のフィールドがあります。この 12 個のフィールドのあとにはオプション フィールドが、さらにそのあとに、通常はトランスポート レイヤ パケットであるデータ部分が続く場合があります。可変長のオプション フィールドは、IPv4 パケット ヘッダーの合計サイズに加算されます。IPv4 パケット ヘッダーのグレーの部分のフィールドは、IPv6 パケット ヘッダーに含まれません。

図 *10 : IPv4* パケット ヘッダー形式



簡易 IPv6 パケット ヘッダー

base IPv6 パケット ヘッダーには、合計サイズが 40 オクテット（320 ビット）の 8 のフィールドがあります。フラグメンテーションはパケットの送信元により処理され、データリンク層のチェックサムとトランスポート層が使用されます。ユーザデータグラムプロトコル（UDP）チェックサムにより、内部パケットと基本 IPv6 パケット ヘッダーの整合性がチェックされ、オプション フィールドが 64 ビットに揃えられるため、IPv6 パケットの処理が容易になります。

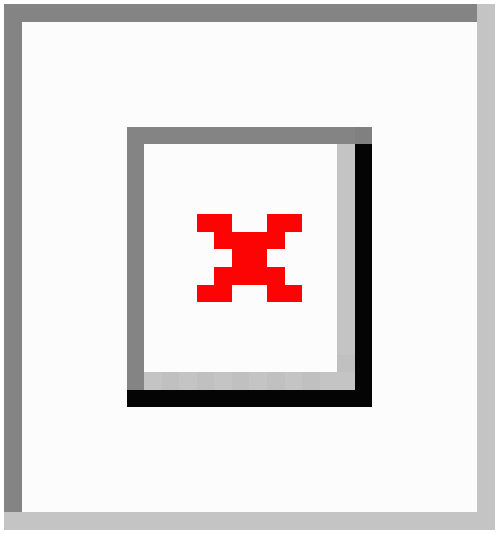
次の表に、基本 IPv6 パケット ヘッダーのフィールドをリストします。

表 *8 : base IPv6* パケット ヘッダー フィールド

フィールド	説明
バージョン	IPv4 パケット ヘッダーのバージョン フィールドに該当しますが、IPv4 で示される数字 4 の代わりに、IPv6 では数字 6 が示されます。
トラフィック クラス	IPv4 パケット ヘッダーのタイプ オブ サービス フィールドと同様です。トラフィック クラス フィールドは、差別化されたサービスで使用されるトラフィック クラスのタグをパケットに付けます。
フロー ラベル	IPv6 パケット ヘッダーの新規フィールドです。フロー ラベル フィールドは、ネットワーク層でパケットを差別化する特定のフローのタグをパケットに付けます。
ペイロード長	IPv4 パケット ヘッダーの合計長フィールドと同様です。ペイロード長フィールドは、パケットのデータ部分の合計長を示します。

フィールド	説明
次ヘッダー	IPv4 パケット ヘッダーのプロトコル フィールドと同様です。次ヘッダー フィールドの値により、基本 IPv6 ヘッダーに続く情報のタイプが決まります。基本 IPv6 ヘッダーに続く情報のタイプは、下の図に示すように、TCP パケット、UDP パケット、または拡張ヘッダーなどのトランスポート層パケットです。
ホップ リミット	IPv4 パケット ヘッダーの存続可能時間フィールドと同様です。ホップ リミット フィールドの値は、IPv6 パケットが無効と見なされる前に通過できるルータの最大数です。各ルータを通過するたびに、この値が 1 つずつ減少します。IPv6 ヘッダーにはチェックサムがないため、ルータは値を減らすたびにチェックサムを再計算する必要がなく、処理リソースが節約されます。
送信元アドレス	IPv4 パケット ヘッダーの送信元アドレス フィールドと同様ですが、IPv4 の 32 ビット送信元アドレスの代わりに、IPv6 では 128 ビットの送信元アドレスが含まれます。
宛先アドレス	IPv4 パケット ヘッダーの宛先アドレス フィールドと同様ですが、IPv4 の 32 ビット宛先アドレスの代わりに、IPv6 では 128 ビットの宛先アドレスが含まれます。

図 *11 : IPv6* パケット ヘッダー形式

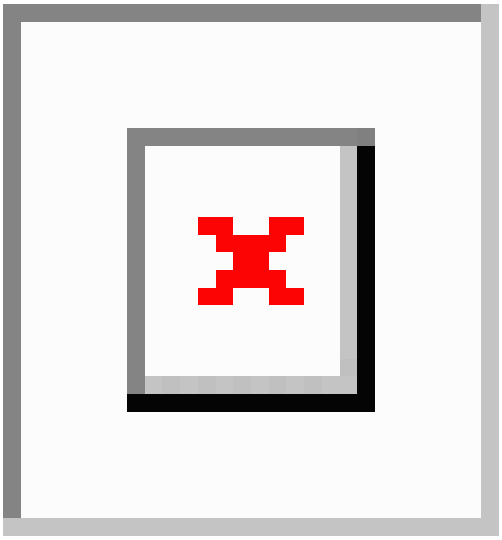


IPv6 拡張ヘッダー

任意に使用できる拡張ヘッダーおよびパケットのデータ部分は、基本 IPv6 パケット ヘッダーの 8 つのフィールドのあとに続きます。存在する場合は、各拡張ヘッダーが 64 ビットに揃えられます。IPv6 パケットの拡張ヘッダーの数は固定されていません。各拡張ヘッダーは、前のヘッダー

の次ヘッダー フィールドによって識別されます。通常は、最後の拡張ヘッダーに、TCP や UDP などのトランスポートレイヤプロトコルの次ヘッダーフィールドがあります。次の図は、IPv6 拡張ヘッダーの形式を示しています。

図 *12 : IPv6* 拡張ヘッダー形式



下表に、拡張ヘッダー タイプとその次ヘッダー フィールド値をリストします。

表 *9 : IPv6* 拡張ヘッダー タイプ

ヘッダー タイプ	次ヘッダーの値	説明
ホップバイホップ オプション	0	パケットのパス上のすべてのホップで処理されるヘッダー。存在する場合、ホップバイホップ オプション ヘッダーは、常に基本 IPv6 パケット ヘッダーの直後に続きます。
宛先オプション	60	任意のホップバイホップ オプション ヘッダーのあとに続くことのあるヘッダー。このヘッダーは、最終の宛先、およびルーティング ヘッダーで指定された各通過アドレスで処理されます。
ルーティング	43	送信元ルーティングに使用されるヘッダー。
フラグメント	44	送信元が、送信元と宛先の間のパスの最大伝送単位（MTU）より大きいパケットをフラグメント化するときに使用されるヘッダー。フラグメント ヘッダーは、フラグメント化された各パケットで使用されます。
認証	51	パケットのコネクションレス型整合性およびデータ発信元認証を提供するために使用されるヘッダー。

ヘッダー タイプ	次ヘッダーの値	説明
Encapsulation Security Payload	50	このヘッダーに続くすべての情報は暗号化されます。
モビリティ	135	モバイル IPv6 サービスのサポートで使用されるヘッダー。
ホスト識別プロトコル	139	Host Identity Protocol バージョン 2（HIPv2）に使用されるヘッダー。IP マルチホーミングとモバイル コンピューティングをセキュアな方法で実現できるようにします。
シム 6	140	IPマルチホーミングに使用されるヘッダー。これにより、ホストを複数のネットワークに接続できます。
上位レイヤ ヘッダー	6（TCP） 17（UDP）	データ転送のためにパケット内で使用されるヘッダー。2 つの主要なトランスポート プロトコルは TCP と UDP です。

（注） 一部のスイッチモデルは、IPv6 拡張ヘッダー タイプのサブセットのみをサポートします。次のリストに、Cisco Nexus 3600 プラットフォームスイッチ（N3K-C36180YC-R および N3K-C3636C-R）、および N9K-X9636Q-R、N9K-X9636C-RX、および N9K-X96136YC-R ライン カードを搭載した Cisco Nexus 9504 および 9508 モジュラ シャーシでサポートされる拡張ヘッダー タイプを示します。。

サポート対象：宛先オプション（60）、ルーティング（43）、フラグメント（44）、モビリティ（135）、ホスト アイデンティティ プロトコル（HIP）（139）、シム 6（140）。

サポート対象外：ホップバイホップ オプション（0）、カプセル化セキュリティ ペイロード（50）、認証ヘッダー（51）、および試験的ヘッダー（253 および 254）。

Cisco NX-OS リリース 9.3(7) 以降では、ここにリストされているデバイスで IPv6 ACL を設定する場合、拡張ヘッダーを含む IPv6 パケットの処理に関する新しいルールを含める必要があります。必要な設定手順については、NX-OS リリース 9.3(x) 以降の『*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*』の「Configuring an ACL for IPv6 Extension Headers」を参照してください。

IPv6 の DNS

IPv6 では、DNS の名前からアドレスおよびアドレスから名前のルックアップ プロセスでサポートされる DNS レコード タイプがサポートされます。DNS レコード タイプは IPv6 アドレスをサポートしています（表を参照）。

（注） IPv6 では、IPv6 アドレスから DNS 名への逆マッピングもサポートされます。

表 *10 : IPv6 DNS* レコード タイプ

レコード タイプ	説明	フォーマット（**Format**）
AAAA	ホスト名を IPv6 アドレスにマッピングします（IPv4 の A レコードと同等）。	www.abc.test AAAA 3FFE:YYYY:C18:1::2
PTR	IPv6 アドレスをホスト名にマッピングします（IPv4 の PTR レコードと同等）。	2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.8.1.c.0.y.y.y.y.e.f.f.3.ip6.int PTR www.abc.test

IPv6 のパス MTU ディスカバリ

IPv4 の場合と同様に、ホストがダイナミックに、データ パス上のすべてのリンクの MTU サイズの差を検出し、それに合わせて調整できるよう、IPv6 でパス MTU ディスカバリを使用できます。ただし、IPv6 では、特定のデータ パス上の 1 つのリンクのパス MTU がパケットのサイズに十分に対応できる大きさでない場合に、フラグメンテーションはパケットの送信元によって処理されます。IPv6 ホストでパケット フラグメンテーションを処理すると、IPv6 ルータの処理リソースが節約され、IPv6 ネットワークの効率が向上します。ICMP の Too Big メッセージの到着によってパス MTU が削減されると、Cisco NX-OS はその低い値を保持します。この接続では、スループットを測定するためにセグメント サイズが増加することはありません。

（注） IPv6 では、最小リンク MTU は 1280 オクテットです。IPv6 リンクには、1500 オクテットの MTU 値の使用を推奨します。

CDP IPv6 アドレスのサポート

ネイバー情報機能用の Cisco Discovery Protocol（CDP）IPv6 アドレスのサポートを使用して、2 台のシスコ デバイス間で IPv6 アドレス指定情報を転送できます。IPv6 アドレス向け Cisco Discovery Protocol サポートは、ネットワーク管理製品およびトラブルシューティング ツールに IPv6 情報を提供します。

IPv6 の ICMP

IPv6 の ICMP を使用して、ネットワークのヘルスの情報について説明します。IPv6 で動作する ICMPv6 は、パケットを正しく処理できない場合にエラーを報告し、ネットワークのステータスに関する情報メッセージを送信します。たとえば、パケットが大きすぎて別のネットワークに送信

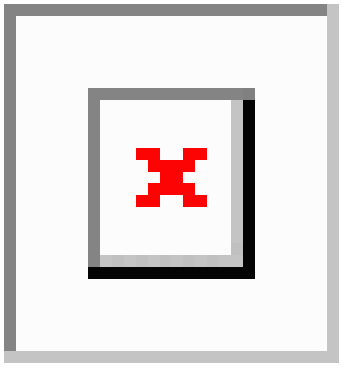
できないためにルータがパケットを転送できない場合、ルータは発信元ホストにICMPv6メッセージを送信します。さらに、IPv6 の ICMP パケットは IPv6 ネイバー探索およびパス MTU ディスカバリに使用されます。パス MTU ディスカバリ プロセスでは、特定のルートでサポートされる最大サイズを使用してパケットが送信されます。

基本 IPv6 パケット ヘッダーの次ヘッダー フィールドの値が 58 の場合は、IPv6 ICMP パケットであることを意味します。ICMP パケットは、すべての拡張ヘッダーの後に続き、IPv6 パケットの最後の情報です。IPv6 ICMP パケット内の [ICMPv6 タイプ（ICMPv6 Type）] フィールドと [ICMPv6 コード（ICMPv6 Code）] フィールドは、ICMP メッセージ タイプなどの IPv6 ICMP パケットの詳細を識別します。[チェックサム（ Checksum ）] フィールドの値は、送信側で計算され、IPv6 ICMP パケットと IPv6 疑似ヘッダーのフィールドから受信者によって確認されます。

（注） IPv6 ヘッダーにはチェックサムがありません。ただし、トランスポート層上のチェックサムにより、パケットが正しく配信されていないかどうかを判定できます。計算に IP アドレスを含むすべてのチェックサム計算は、新しい 128 ビット アドレスに対応するように IPv6 用に変更する必要があります。チェックサムは、疑似ヘッダーを使用して生成されます。

ICMPv6 ペイロード フィールドには、IP パケット処理に関連するエラー情報または診断情報が含まれます。次の図は、IPv6 ICMP パケット ヘッダーの形式を示しています。

図 *13 : IPv6 ICMP* パケット ヘッダーの形式



IPv6 ネイバー探索

IPv6 ネイバー探索プロトコル（NDP）を使用して、ネイバー ルータが到達可能かどうかを判断できます。IPv6 ノードは、ネイバー探索を使用して、同じネットワーク（ローカル リンク）上のノードのアドレスを決定し、パケットを転送できるネイバー ルータを見つけ、ネイバー ルータが到達可能かどうかを確認し、リンク層アドレスの変更を検出します。NDP は、ICMP メッセージを使用して、到達不能な隣接ルータにパケットが送信されたかどうかを検出します。

IPv6 ネイバー送信要求メッセージ

ノードは、同じローカル リンク上の別のノードのリンク層アドレスを決定するときに、ICMP パケット ヘッダーのタイプ フィールドの値が 135 であるネイバー送信要求メッセージをローカル リンクで送信します（下記の図を参照）。送信元アドレスは、ネイバー送信要求メッセージを送信するノードの IPv6 アドレスです。宛先アドレスは、宛先ノードの IPv6 アドレスに対応する送信要求ノードマルチキャストアドレスです。ネイバー送信要求メッセージには、送信元ノードのリンク層アドレスも含まれます。

図 *14 : IPv6* ネイバー探索 - ネイバー送信要求メッセージ

ネイバー送信要求メッセージを受信した後に、宛先ノードは、ICMP パケット ヘッダーのタイプ フィールドに値 136 を含むネイバー アドバタイズメント メッセージをローカル リンクに送信することで応答します。送信元アドレスは、ネイバー アドバタイズメント メッセージを送信するノードの IPv6 アドレス（ノード インターフェイスの IPv6 アドレス）です。宛先アドレスは、ネイバー送信要求メッセージを送信するノードのIPv6アドレスです。データ部分には、ネイバー アドバタイズメント メッセージを送信するノードのリンク層アドレスが含まれます。

送信元ノードがネイバー アドバタイズメントを受信すると、送信元ノードと宛先ノードが通信できるようになります。

ネイバー送信要求メッセージは、ノードがネイバーのリンク層アドレスを識別した後に、ネイバーの到達可能性を確認できます。ノードがあるネイバーの到達可能性を検証する場合、そのネイバーのユニキャスト アドレスとして、ネイバー送信要求メッセージ内の宛先アドレスを使用します。

ネイバー アドバタイズメント メッセージは、ローカル リンク上のノードのリンク層アドレスが変更されたときにも送信されます。変更があった場合、ネイバー アドバタイズメントの宛先アドレスは全ノード マルチキャスト アドレスになります。

ネイバー到達不能検出では、ネイバーの障害またはネイバーへの転送パスの障害が識別されます。この検出は、ホストとネイバー ノード（ホストまたはルータ）間のすべてのパスで使用されます。ネイバー到達不能検出は、ユニキャスト パケットだけが送信されるネイバーに対して実行され、マルチキャスト パケットが送信されるネイバーに対しては実行されません。

ネイバーは、（以前にネイバーに送信されたパケットが受信され、処理されたことを示す）肯定確認応答がネイバーから返された場合に、到達可能と見なされます。肯定確認応答（TCP などの上位層プロトコルからの）は、接続が順調に進んでいる（宛先に到達しつつある）ことを示します。パケットがピアに到達している場合、それらのパケットは送信元のネクストホップ ネイバーにも到達しています。転送の進行により、ネクストホップ ネイバーが到達可能であることも確認されます。

ローカルリンク上にない宛先の場合、転送の進行は、ファーストホップルータが到達可能であることを暗に意味します。上位層プロトコルからの確認応答がない場合、ノードは、ユニキャスト ネイバー送信要求メッセージを使用してネイバーを探し、転送パスがまだ機能していることを確認します。ネイバーから返信された請求ネイバーアドバタイズメントメッセージは、転送パスがまだ機能しているという肯定確認応答です（請求フラグが値 1 に設定されたネイバー アドバタイズメント メッセージは、ネイバー請求メッセージへの返信としてだけ送信されます）。非送信要

求メッセージでは、送信元ノードから宛先ノードへの一方向パスだけが確認されます。送信要求ネイバー アドバタイズメント メッセージは、両方向のパスが機能していることを示します。

（注） 送信要求フラグが値 0 に設定されたネイバー アドバタイズメント メッセージは、転送パスがまだ機能していることを示す肯定確認応答とは見なされません。

ネイバー送信要求メッセージは、ユニキャスト IPv6 アドレスがインターフェイスに割り当てられる前にそのアドレスが一意であることを確認するために、ステートレス自動設定プロセスでも使用されます。新規のリンクローカル IPv6 アドレスに対しては、アドレスがインターフェイスに割り当てられる前に、最初に重複アドレス検出が実行されます（重複アドレス検出の実行中、新規アドレスは一時的な状態のままです）。ノードは未指定の送信元アドレスと一時的なリンクローカル アドレスをメッセージの本文に含むネイバー送信要求メッセージを送信します。そのアドレスが別のノードですでに使用されている場合、ノードは一時的なリンクローカル アドレスを含むネイバーアドバタイズメントメッセージを返します。別のノードが同じアドレスの一意性を同時に検証している場合は、そのノードもネイバー送信要求メッセージを返します。ネイバー送信要求メッセージの返信としてネイバーアドバタイズメントメッセージが受信されず、同じ一時アドレスの検証を試行している他のノードからのネイバー送信要求メッセージも受信されない場合、最初のネイバー送信要求メッセージを送信したノードは、一時的なリンクローカル アドレスを一意であると見なし、そのアドレスをインターフェイスに割り当てます。

IPv6 ステートレス自動設定

IPv6 ノードのすべてのインターフェイスは、インターフェイスの ID およびリンクローカル プレフィックス FE80::/10 から自動的に設定されるリンクローカル アドレスを持つ必要があります。リンクローカル アドレスを使用すると、ノードがリンク上の他のノードと通信できます。また、リンクローカル アドレスを使用して、ノードをさらに設定することもできます。

IPv6 ステートレスアドレス自動構成（SLAAC）は、管理インターフェイスでのみ実行されます。たとえば、SLAAC が管理インターフェイスで有効になっている場合、リンク ローカル アドレス（LLA）が生成され、リンク ローカルアドレスに対して重複アドレス検出（DAD）が実行されます。重複アドレス検出プロセスが成功すると、インターフェイスは ICMPv6 ルータ要請（RS）パケットを送信します。RS パケットを受信するアップストリーム ルータは、ICMPv6 ルータ アドバタイズメント（RA）で応答します。RA パケットには、インターフェイスの MAC 情報と RA パケット内のアドバタイズされたプレフィックスを使用して、ダウンストリーム NX-OS スイッチがアドレスを自動生成するサブネットを伝送するプレフィックス TLV オプションがあります。Cisco NX-OS スイッチは、EUI-64 形式でアドレスを自動生成し、新しい自動生成されたアドレスで DAD を実行します。

IPv6 アドレスは、特定の時間だけインターフェイスに割り当てられます。各アドレスには、アドレスがインターフェイスに接続されている期間を示すライフタイムがあります。アップストリーム ルータから送信される RA パケットの TLV プレフィックスには、有効なライフタイムと優先ライフタイムに関する情報が含まれています。インターフェイスに割り当てられたアドレスは、2 つの異なるフェーズを通過します。最初は、アドレスは優先状態になります。これは、アドレスが任意の通信での使用に制限されないことを意味します。現在のインターフェイス バインディン

グが無効になると、アドレスは廃止状態になります。廃止状態では、アドレスの使用は推奨されません。必ずしも禁止されているわけではありません。サービスを中断せずに別のアドレスに切り替えることが困難なアプリケーションのみが、廃止されたアドレスを使用する必要があります。

IPv6 コンピューティング ノード IP 自動構成

K8s クラスタにオンボードし、スイッチとコンピューティング ノード間で eBGP ピアリングを確立する前に、接続されたコンピューティング ノードをノードIPを接続されたコンピューティングノードに割り当てる必要があります。

Cisco NX-OS リリース 10.3(3)F 以降では、Cisco NX-OS 9000 シリーズ プラットフォーム スイッチで IPv6 コンピューティング ノード IP 自動構成のサポートが提供され、マルチホーム コンピューティング ノードにノード IP アドレスを割り当てて配布し、割り当てられたノードIPを使用して K8s クラスタに到達可能性を確立します。

（注） ただし、ノードアドレスの割り当てはSLAACとは異なります。これは、SLAACを介して実行されるレイヤ 3 インターフェイス サブネットでのインターフェイス アドレス プロビジョニングと直交する、ループバック インターフェイスに固有の IPv6 アドレスを割り当てる方法です。

この機能は、8505/6775 で定義されている標準に準拠しています。

IPv6 ルータ アドバタイズメント メッセージ

ルータ アドバタイズメント（RA）メッセージは、ICMP パケット ヘッダーのタイプ フィールドが値 134 であり、IPv6 ルータの構成済みの各インターフェイスへ定期的に送信されます。ステートレス自動構成が正しく機能するには、RA メッセージでアドバタイズされたプレフィックス長が常に 64 ビットである必要があります。

RA メッセージは、全ノード マルチキャスト アドレスに送信されます（以下の図を参照）。

図 *15 : IPv6* ネイバー検出：*RA* メッセージ

RA メッセージは、全ノード マルチキャスト アドレスに送信されます。

RA メッセージには、通常次の情報が含まれています。

- ローカル リンク上のノードがその IPv6 アドレスの自動設定に使用できる 1 つ以上のオンリンク IPv6 プレフィックス
- アドバタイズメントに含まれる各プレフィックスのライフタイム情報
- 完成可能な自動設定のタイプ（ステートレスまたはステートフル）を示すフラグのセット

- デフォルトルータ情報（アドバタイズメントを送信しているルータをデフォルトルータとして使用する必要があるかどうか、および、その場合は、ルータがデフォルト ルータとして使用される秒単位の時間）
- ホストが発信するパケットで使用する必要のあるホップ リミットや MTU など、ホストに関する詳細情報

RAは、ルータ送信要求メッセージへの応答としても送信されます。ICMPパケットヘッダーのタイプ フィールドの値が 133 であるルータ送信要求メッセージは、システム始動時にホストによって送信されるため、ホストは次のスケジュールされた RA メッセージを待機することなくすぐに自動設定できます。通常、送信元アドレスは未指定の IPv6 アドレス（0:0:0:0:0:0:0:0）です。ホストに設定済みのユニキャスト アドレスがある場合、ルータ送信要求メッセージを送信するインターフェイスのユニキャスト アドレスが、メッセージ内の送信元アドレスとして使用されます。宛先アドレスは、リンク範囲の全ルータ マルチキャスト アドレス。ルータ送信要求に応答して RA が送信される場合、RA メッセージ内の宛先アドレスはルータ送信要求メッセージの送信元のユニキャスト アドレスです。

次の RA メッセージ パラメータを構成できます。

- RA メッセージの定期的な時間間隔
- （特定のリンク上のすべてのノードで使用される）デフォルト ルータとしてのルータの実用性を示す「ルータ ライフタイム」値
- 特定のリンクで使用されているネットワーク プレフィックス
- （特定のリンクで）ネイバー送信要求メッセージが再送信される時間の間隔
- ノードによってネイバーが到達可能である（特定のリンク上のすべてのノードで使用できる）と見なされるまでの時間

設定されたパラメータはインターフェイスに固有です。RAメッセージ（デフォルト値を含む）の送信は、イーサネットとインターフェイス上では自動的にイネーブルになります。他のインターフェイス タイプの場合は、**no ipv6 nd suppress-ra** コマンドを入力して RA メッセージを送信する必要があります。個々のインターフェイスでは、**ipv6 nd suppress-ra** コマンドを入力して、RA メッセージ機能を無効にできます。

IPv6 ネイバー リダイレクト メッセージ

ルータは、ネイバーリダイレクトメッセージを送信して、宛先へのパス上のより適切なファーストホップ ノードをホストに通知しますICMP パケット ヘッダーのタイプ フィールドの値 137 は、IPv6 ネイバー リダイレクト メッセージを示します。

図 *16 : IPv6* ネイバー探索：ネイバー リダイレクト メッセージ

（注） リダイレクトメッセージ内のターゲットアドレス（最終的な宛先）によって隣接ルータのリンクローカルアドレスが確実に識別されるように、ルータは各隣接ルータのリンクローカルアドレスを判断できる必要があります。静的ルーティングの場合は、ルータのリンクローカル アドレスを使用してネクストホップ ルータのアドレスを指定する必要があります。動的ルーティングの場合、隣接ルータのリンクローカル アドレスを交換するように、すべての IPv6 ルーティング プロトコルを構成する必要があります。

パケットの転送後に、次の条件が満たされる場合、ルータはパケットの送信元にリダイレクトメッセージを送信します。

- パケットの宛先アドレスがマルチキャスト アドレスではない。
- パケットがルータにアドレッシングされていなかった。
- パケットが、そのパケットを受信したインターフェイスから送信されようとしている。
- ルータが、パケットにより適したファーストホップ ノードはパケットの送信元と同じリンク上にあると判断した。
- パケットの送信元アドレスが、同じリンク上のネイバーのグローバル IPv6 アドレス、またはリンクローカル アドレスである。

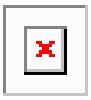
IPv6 エニーキャスト アドレス

エニーキャスト アドレスとは、異なるノードに属するインターフェイス一式に割り当てられたアドレスです。エニーキャストアドレスに送信されたパケットは、使用しているルーティングプロトコルの定義に従って、そのエニーキャスト アドレスが示す最も近いインターフェイスに送信されます。エニーキャストアドレスは、ユニキャストアドレス空間から割り当てられるため、その構文ではユニキャストアドレスと区別できません。ユニキャストアドレスを複数のインターフェイスに割り当てると、ユニキャスト アドレスがエニーキャスト アドレスとなります。属するエニーキャストアドレスが割り当てられたノードは、アドレスがエニーキャストアドレスであることを認識できるよう、設定する必要があります。

（注） エニーキャストアドレスを使用できるのは、ルータだけです。ホストはエニーキャストアドレスを使用できません。エニーキャスト アドレスは、IPv6 パケットの送信元アドレスには使用できません。

次の図は、サブネット ルータ エニーキャスト アドレスのフォーマットを示します。このアドレスには、連続するゼロに連結されたプレフィックス（インターフェイス ID）があります。サブネット ルータ エニーキャスト アドレスを使用すると、サブネット ルータ エニーキャスト アドレスのプレフィックスが示すリンク上のルータに到達できます。

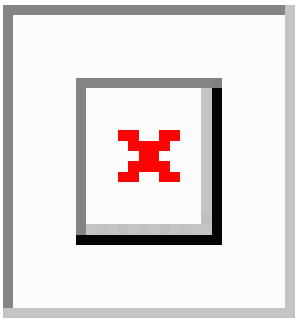
図 *17 :* サブネット ルータ エニーキャスト アドレスの形式



IPv6 マルチキャスト アドレス

IPv6 マルチキャスト アドレスは、FF00::/8（1111 1111）というプレフィックスを持つ IPv6 アドレスです。IPv6 マルチキャスト アドレスは、異なるノードに属するインターフェイス一式の ID です。マルチキャストアドレスに送信されたパケットは、マルチキャストアドレスが示すすべてのインターフェイスに配信されます。プレフィックスに続く 2 番めのオクテットで、マルチキャストアドレスのライフタイムとスコープが定義されます。永久マルチキャストアドレスはライフタイム パラメータが 0 に等しく、一時マルチキャスト アドレスのライフタイム パラメータは 1 に等しくなっています。ノード、リンク、サイト、または組織のスコープ、またはグローバル スコープを持つマルチキャスト アドレスのスコープ パラメータはそれぞれ、1、2、5、8、または E です。たとえば、プレフィックスが FF02::/16 のマルチキャスト アドレスは、リンク スコープを持つ永続マルチキャストアドレスです。次の図に、IPv6マルチキャストアドレスの形式を示します。

図 *18 : IPv6* マルチキャスト アドレス形式



IPv6 ノード（ホストとルータ）は、（受信パケットの宛先となる）次のマルチキャスト グループに加入する必要があります。

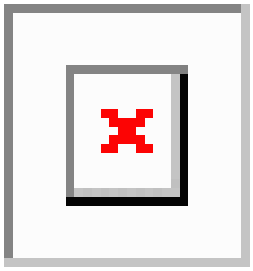
- 全ノード マルチキャスト グループ FF02:0:0:0:0:0:0:1（スコープはリンクローカル）
- 割り当てられたユニキャストアドレスおよびエニーキャストアドレスごとの送信要求ノード マルチキャスト グループ FF02:0:0:0:0:1:FF00:0000/104

IPv6 ルータは、全ルータ マルチキャスト グループ FF02:0:0:0:0:0:0:2（スコープはリンクローカル）にも加入する必要があります。

送信要求ノード マルチキャスト アドレスは、IPv6 ユニキャスト アドレスまたはエニーキャスト アドレスに対応するマルチキャスト グループです。IPv6 ノードは、割り当てられているユニキャストアドレスおよびエニーキャストアドレスごとに、関連付けられた送信要求ノードマルチキャスト グループに加入する必要があります。IPv6 送信要求ノード マルチキャスト アドレスには、対応する IPv6 ユニキャスト アドレスまたは IPv6 エニーキャスト アドレスの下位 24 ビットに連結されたプレフィックス FF02:0:0:0:0:1:FF00:0000/104 があります（下図を参照）。たとえば、IPv6

アドレス 2037::01:800:200E:8C6C に対応する送信要求ノード マルチキャスト アドレスは FF02::1:FF0E:8C6C です。送信要求ノード アドレスは、ネイバー送信要求メッセージで使用されます。

図 *19 : IPv6* 送信要求ノード マルチキャスト アドレス形式



（注） IPv6 にはブロードキャスト アドレスはありません。ブロードキャスト アドレスの代わりに IPv6 マルチキャスト アドレスが使用されます。

LPMルーティングモード

デフォルトでは、Cisco NX-OSは、デバイス上で最長プレフィックス一致（LPM）を許可するように階層的にルーティングします。ただし、より多くの LPM ルート エントリをサポートするために、異なるルーティング モード用にデバイスを設定できます。

次の表に、Cisco Nexus 9300 シリーズおよび 9500 シリーズ スイッチでサポートされている LPM ルーティング モードを示します。

表 *11 : Cisco Nexus 9200* シリーズ スイッチ用の *LPM* ルーティング モード

LPM ルーティング モード	CLI コマンド
デフォルトのシステム ルーティング モード	
LPM デュアルホスト ルーティング モード	**system routing template-dual-stack-host-scale**
LPM ヘビー ルーティング モード	**system routing template-lpm-heavy**

（注） Cisco Nexus 9200 プラットフォーム スイッチは、IPv4 マルチキャスト ルートの **system routing template-lpm-heavy** モードをサポートしていません。LPM の上限を 0 にリセットしてください。

表 *12 : Cisco Nexus 9300* シリーズ スイッチ用の *LPM* ルーティング モード

LPM ルーティング モード	Broadcom T2モード	CLI コマンド
デフォルトのシステム ルーティング モード	3	
ALPM ルーティング モード	4	**system routing max-mode l3**

表 *13 : Cisco Nexus 9300-EX/FX/FX2/FX3/GX* プラットフォーム スイッチ用の *LPM* ルーティング モード

LPM ルーティング モード	CLI コマンド
LPM デュアルホスト ルーティング モード	**system routing template-dual-stack-host-scale**
LPM ヘビー ルーティング モード	**system routing template-lpm-heavy**
LPM インターネットピアリング モード)	**system routing template-internet-peering**

表 *14 : 9700-EX* および *9700-FX* ラインカ ードを搭載した *Cisco Nexus 9500* プラットフォーム スイッチ用 *LPM* ルーティング モード

LPM ルーティング モード	Broadcom T2モード	CLI コマンド
デフォルトのシステムルーティング モード	3（ラインカード用）。 4（ファブリック モジュール用）	
最大-ホスト ルーティング モード	2（ラインカード用）。 3（ファブリック モジュール用）	**system routing max-mode host**
非階層ルーティングモード	3（ラインカード用）。 max-l3-modeオプション付き4（ラインカード用）	**system routing non-hierarchical-routing [max-l3-mode]**
64 ビット ALPM ルーティング モード	モード4のサブモード（ファブリックモジュール用）	**system routing mode hierarchical 64b-alpm**

LPM ルーティング モード	Broadcom T2モード	CLI コマンド
LPM ヘビー ルーティング モード		**system routing template-lpm-heavy** （注） このモードは、9732C-EX ライン カードを搭載した Cisco Nexus 9508 スイッチでのみサポートされます。
LPMインターネットピアリング モード)		**system routing template-internet-peering** （注） このモードは、次の Cisco Nexus 9500 プラットフォーム スイッチでのみサポートされています。 • 9700-EX ライン カード搭載の Cisco Nexus 9500 プラットフォーム スイッチ • Cisco Nexus 9500-FX プラットフォーム スイッチ（Cisco NX-OS リリース 7.0(3)I7(4) 以降） • Cisco 9500-R プラットフォーム スイッチ（Cisco NX-OS リリース 9.3(1) 以降）
LPM デュアルホスト ルーティング モード		

表 *15 : 9600-R* ライン カードを搭載した *Cisco Nexus 9500-R* プラットフォーム スイッチの *LPM* ルーティング モード

LPM ルーティング モード	CLI コマンド
LPMインターネットピアリング モード)	**system routing template-internet-peering** （Cisco NX-OS リリース 9.3(1) 以降 ）

ホストから LPM へのスピルオーバー

Cisco NX-OS リリース 7.0(3)I5(1) 以降では、ホストルートを LPM テーブルに保存して、より大きなホスト スケールを実現できます。ALPM モードでは、スイッチはより少ないホスト ルートを許可します。サポートされるスケールよりも多くのホストルートを追加すると、ホストテーブルからこぼれたルートは LPM テーブルの LPM ルートのスペースを使用します。このモードで許可される LPM ルートの総数は、保存されているホスト ルートの数だけ減少します。この機能は、Cisco Nexus 9300 および 9300 プラットフォーム スイッチではサポートされていません。

デフォルトのシステム ルーティング モードでは、Cisco Nexus 9300 プラットフォーム スイッチは、より高いホスト スケールとより少ない LPM ルート用に設定され、より多くのホスト ルートを保存するために LPM スペースを使用できます。Cisco Nexus 9500 プラットフォーム スイッチでは、デフォルトのシステム ルーティング モードと非階層型ルーティング モードのみがライン カードでこの機能をサポートします。ファブリック モジュールはこの機能をサポートしていません。

仮想化のサポート

IPv6 は、仮想ルーティング/転送（VRF）インスタンスをサポートします。

ECMP を使用した IPv6 ルート

ルートのすべてのネクストホップが収集、ドロップ、またはパントの場合、すべてのネクストホップはマルチパス ハードウェア テーブルにそのままプログラムされます。

ルートの一部のネクストホップがグリーニング、ドロップ、またはパントであり、残りのネクストホップがそうでない場合、非グリーニング、ドロップ、またはパントのネクストホップのみがマルチパス ハードウェア テーブルにプログラムされます。

ECMP ルートの特定のネクストホップが解決されると（ARP/IPV6 ND が解決されると）、それに応じてマルチパス ハードウェア テーブルが更新されます。

IPv6の前提条件

IPv6 には、次の前提条件があります。

- IPv6 アドレッシングおよび IPv6 ヘッダー情報などの IPv6 の基本に関する詳しい知識が必要です。
- デバイスをデュアルスタック デバイス（IPv4/IPv6）にする場合は、必ずメモリ/処理の注意事項に従ってください。

IPv6 の注意事項および制約事項

IPv6 設定時の注意事項および制約事項は、次のとおりです。

- インターネット ピアリング モードに設定された Cisco Nexus 9300-EX および Cisco Nexus 9300-FX2 プラットフォーム スイッチには、完全な IPv4 および IPv6 インターネット ルートを同時にインストールするための十分なハードウェア容量がない場合があります。
- スイッチは、IPv6 フレームを転送する前にレイヤ 3 パケット情報を確認しないため、IPv6 パケットは、レイヤ 2 LAN スイッチに対して透過的です。IPv6 ホストは、レイヤ 2 LAN スイッチに直接接続できます。

- インターフェイスの同じプレフィックス内に複数の IPv6 グローバル アドレスを設定できます。ただし、1 つのインターフェイス上での複数の IPv6 リンクローカル アドレスはサポートされません。
- IPv6 LLA を使用するには、 **ing-sup** の TCAM リージョンをデフォルト値の 512 から 768 に再分割する必要があります。この手順では、コピー実行の開始とリロードが必要です。
- IPv6 スタティック ルートのネクストホップ リンクローカル アドレスは、どのローカル インターフェイスでも設定できません。
- リンク ローカル IPv6 アドレスを使用する場合は、BGP 更新ソースを定義する必要があります。
- RFC 3879 によりサイトローカル アドレスの使用が廃止されたため、RFC 4193 のユニーク ローカル アドレス（UCA）の推奨に従って、プライベート IPv6 アドレスを設定する必要があります。
- Cisco Nexus 9500-R プラットフォーム スイッチの場合、インターネット ピアリング モードは、グローバル インターネット ルーティング テーブルで配信されるプレフィックス パターンでのみ使用されます。このモードでは、他のプレフィックス配布/パターンは動作できますが、予測できません。その結果、プレフィックスパターンが実際のインターネットプレフィックス パターンである場合にのみ、達成可能な最大 LPM/LEM スケールが信頼できます。インターネット ピアリング モードでは、グローバル インターネット ルーティング テーブル内のルート プレフィックス パターン以外のルート プレフィックス パターンが使用されている場合、スイッチは文書化されたスケーラビリティの数値を正常に達成できない可能性があります。
- LPM の重いルーティングモードは、**9700**-EX、-FX、および-GX シリーズモジュールを搭載した Cisco Nexus **9500** シリーズスイッチでサポートされます。
- Cisco NX-OS リリース 10.2(3)F 以降、設定された間隔に基づいて IPv6 リダイレクト メッセージがトリガーされると、syslog が出力されます。
- Cisco NX-OS リリース 10.3(1)F 以降、スタティック ルーティングが Cisco Nexus 9808 スイッチでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、静的ルーティングが Cisco Nexus 9804 スイッチでサポートされます。
- Cisco NX-OS リリース 10.3(1)F 以降、ダイナミック ルーティングが Cisco Nexus 9808 スイッチでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、ダイナミック ルーティングが Cisco Nexus 9804 スイッチでサポートされます。
- Cisco NX-OS リリース 10.3(3)F 以降、IPv6 コンピューティング ノード IP 自動構成機能は、次の制限付きで Cisco NX-OS 9000 シリーズ プラットフォーム スイッチでサポートされます。
    - RA プレフィックスは、プレフィックス長が 64 のオフリンクとして構成する必要があります。

- マルチホーム コンピューティング ノードがある場合は、L1 スイッチと L2 スイッチの両方で同じ RA プレフィックスを構成する必要があります。

- Cisco NX-OS リリース 10.4(1)F 以降、ダイナミック ルーティングは、9808 および 9804 スイッチを搭載した Cisco Nexus X98900CD-A および X9836DM-A ライン カードでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、スタティック ルーティングは、9808 および 9804 スイッチを搭載した Cisco Nexus X98900CD-A および X9836DM-A ライン カードでサポートされます。

IPv6 の設定

IPv6 アドレッシングの設定

インターフェイスの IPv6 アドレスを設定して、インターフェイスが IPv6 トラフィックを転送できるようにします。インターフェイスでグローバル IPv6 アドレスを設定すると、リンクローカルアドレスが自動的に設定され、そのインターフェイスで IPv6 が有効となります。

手順の概要

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

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**interface ethernet** *number*
	例：
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

	コマンドまたはアクション	目的

Step 3	**ipv6 address** {*address* [**eui64**] [**route-preference** *preference*] [**secondary**] [**tag** *tag-id*] or **ipv6 address** *ipv6-address* **use-link-local-only**
	例：
	`switch(config-if)# ipv6 address 2001:0DB8::1/10`
	または
	`switch(config-if)# ipv6 address use-link-local-only`
		インターフェイスに割り当てられている IPv6 アドレスを指定し、そのインターフェイスで IPv6 処理をイネーブルにします。
		ipv6 address コマンドを入力すると、IPv6 アドレスの下位 64 ビットにインターフェイス ID を含むグローバル IPv6 アドレスが設定されます。指定する必要があるのはアドレスの 64 ビット ネットワーク プレフィックスだけです。最後の 64 ビットはインターフェイス ID から自動的に計算されます。
		ipv6 address use-link-local-only を入力します。 コマンドを入力すると、インターフェイスのリンクローカルアドレスが設定されます。このアドレスは、IPv6 がインターフェイスでイネーブルになっているときに自動的に設定されるリンクローカル アドレスの代わりに使用されます。
		このコマンドは、IPv6 アドレスを設定せずに、インターフェイス上で IPv6 処理をイネーブルにします。

Step 4	（任意） **show ipv6 interface** 例： `switch(config-if)# show ipv6 interface`	IPv6 用に設定されたインターフェイスを表示します。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

例

次に、IPv6 アドレスを設定する例を示します。

```
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
```

次に、IPv6 インターフェイスを表示する例を示します。

```
switch(config-if)# show ipv6 interface ethernet 3/1
Ethernet3/1, Interface status: protocol-down/link-down/admin-down, iod: 36
     IPv6 address: 2001:db8:0000:0000:0218:baff:fed8:239d
     IPv6 subnet: 2001:db8::/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
```

最大ホスト ルーティング モードの設定（Cisco Nexus 9500 プラットフォーム スイッチのみ）

デフォルトでは、デバイスは階層方式で（モード4になるように設定されたファブリック モジュールとモード 3 になるように設定されたラインカード モジュールで）ルートをプログラミングし、デバイス上での最長プレフィクス照合（LPM）とホスト スケールが可能になります。

デフォルトの LPM およびホスト スケールを変更してシステム内のホストをさらにプログラミングできます。これは、ノードをレイヤ 2 ～ レイヤ 3 の境界ノードとして位置付けるときに必要になる場合があります。

（注） LPM テーブルのエントリをさらに拡大したい場合は、「非階層ルーティング モードの設定（Configuring Nonhierarchical Routing Mode (Cisco Nexus 9500 シリーズ スイッチのみ）」の項を参照して、ライン カード上のレイヤ 3 IPv4 および IPv6 ルートすべてをプログラミングしてファブリック モジュール上のルートはそのままにするようデバイスを設定します。

（注） この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

（注） 最大ホスト ルーティング モードのスケール数については、『Cisco Nexus 9000 シリーズ NX-OS 検証済みスケーラビリティ ガイド』を参照してください。

手順の概要

1. **configure terminal**
2. **[no] system routing max-mode host**
3. （任意） **show forwarding route summary**
4. **copy running-config startup-config**

5. **reload**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	[**no**] **system routing max-mode host** 例： `switch(config)# system routing max-mode host`	ライン カードを Broadcom T2 モード 2 に、ファブリック モジュールを Broadcom T2 モード 3 にして、サポートされるホスト数を増やします。
Step 3	（任意） **show forwarding route summary** 例： `switch(config)# show forwarding route summary`	LPM ルーティング モードを表示します。
Step 4	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。
Step 5	**reload** 例： `switch(config)# reload`	デバイス全体をリブートします。

非階層ルーティング モードの設定（Cisco Nexus 9500 シリーズ スイッチのみ）

ホストの規模が小さい場合（純粋なレイヤ 3 配置の場合など）、コンバージェンス パフォーマンスを向上させるために、ラインカードの最長プレフィクス照合（LPM）のルートをプログラミングすることを推奨します。そうすることによって、ラインカードのルートおよびホストがプログラミングされ、ファブリック モジュールのルートはプログラミングされません。

（注） この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

手順の概要

1. **configure terminal**

2. [**no**] **system routing non-hierarchical-routing** [**max-l3-mode**]
3. （任意） **show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	[**no**] **system routing non-hierarchical-routing** [**max-l3-mode**] 例： `switch(config)# system routing non-hierarchical-routing max-l3-mode`	ラインカードを Broadcom T2モード 3（または **max-l3-mode** オプションを使用している場合は Broadcom T2 モード 4）にし、より大きな LPM スケールをサポートします。その結果、IPv4 および IPv6 ルートのすべてが、ファブリック モジュールではなくラインカードでプログラミングされます。

Step 3	（任意） **show forwarding route summary**
	例：
	`switch(config)# show forwarding route summary`
	`Mode 3:`
	`120K IPv4 Host table`
	`16k LPM table (> 65 < 127 1k entry reserved)`
	`Mode 4:`
	`16k V4 host/4k V6 host`
	`128k v4 LPM/20K V6 LPM`
		LPM モードを表示します。

Step 4	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。
Step 5	**reload** 例： `switch(config)# reload`	デバイス全体をリブートします。

64 ビット ALPM ルーティング モードの設定（Cisco Nexus 9500 プラットフォーム スイッチのみ）

64 ビットアルゴリズム最長プレフィックス一致（ALPM）機能を使用して、IPv4 および IPv6 ルートテーブルエントリを管理できます。64 ビット ALPM ルーティング モードでは、デバイスに保存できるルート エントリの数が増加します。このモードでは、次のいずれかをプログラムできます。

- 80,000 IPv6 エントリ、IPv4 エントリなし
- IPv6 エントリなし、128,000 の IPv4 エントリ
- *x* 個の IPv6 エントリと IPv4 エントリ（$2x + y$ の場合）

（注） この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

（注） 64 ビット ALPM ルーティング モードのスケール数については、『Cisco Nexus 9000 シリーズ NX-OS 検証済みスケーラビリティ ガイド』を参照してください。

手順の概要

1. **configure terminal**
2. **[no] system routing mode hierarchical 64b-alpm**
3. （任意） **show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**[no] system routing mode hierarchical 64b-alpm** 例： `switch(config)# system routing mode hierarchical 64b-alpm`	マスク長が 64 以下のすべての IPv4 および IPv6 LPM ルートをファブリックモジュールにプログラミングします。IPv4 および IPv6 のすべてのホスト ルート、

	コマンドまたはアクション	目的
		およびマスク長が 65 ～ 127 であるすべての LPM ルートがライン カードでプログラミングされます。
Step 3	（任意） **show forwarding route summary** 例： `switch(config)# show forwarding route summary`	LPM モードを表示します。
Step 4	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。
Step 5	**reload** 例： `switch(config)# reload`	デバイス全体をリブートします。

ALPM ルーティング モードの設定（Cisco Nexus 9300 プラットフォーム スイッチのみ）

Cisco Nexus 9300 プラットフォーム スイッチは、多数の LPM ルート エントリをサポートするように設定できます。

（注） この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

（注） ALPM ルーティング モードのスケール数については、『Cisco Nexus 9000 シリーズ NX-OS 検証済みスケーラビリティ ガイド』を参照してください。

手順の概要

1. **configure terminal**
2. **[no] system routing max-mode l3**
3. （任意） **show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例： `switch# configure terminal switch(config)#`	グローバル設定モードを開始します。
Step 2	[**no**] **system routing max-mode l3** 例： `switch(config)# system routing max-mode l3`	デバイスを Broadcom T2 モード 4にして、より大きな LPM スケールをサポートします。
Step 3	（任意） **show forwarding route summary** 例： `switch(config)# show forwarding route summary`	LPM モードを表示します。
Step 4	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。
Step 5	**reload** 例： `switch(config)# reload`	デバイス全体をリブートします。

IPv6 ネイバー探索の構成

ルータで IPv6 ネイバー探索を構成できます。NDP は、IPv6 ノードとルータを有効にして、同じリンク上のネイバーのリンク層アドレスを特定し、隣接ルータを見つけ、ネイバーの動向を把握します。

始める前に

最初にインターフェイスで IPv6 を有効にする必要があります。

手順の概要

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. （任意） **show ip nd interface**
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**interface ethernet** *number*
	例：
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 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**] 例： `switch(config-if)# ipv6 nd prefix`	インターフェイスに割り当てられている IPv6 アドレスを指定し、そのインターフェイスで IPv6 処理をイネーブルにします。 • **hop-limit**：IPv6 ネイバー検出パケットでホップリミットをアドバタイズします。値の範囲は 0 ～ 255 です。 • **managed-config-flag**：ステートフル アドレス自動構成を使用してアドレス情報を取得するために、ICMPv6 ルータ アドバタイズメント メッセージ内でアドバタイズします。 • **mtu**：このリンク上で ICMPv6 ルータ アドバタイズメント メッセージで最大伝送単位（MTU）をアドバタイズします。範囲は 1280 ～ 65535 バイトです。 • **ns-interval**：IPv6 ネイバー送信要求メッセージ間の再送信間隔を構成します。範囲は 1000 ～ 3600000 ミリ秒です。 • **other-config-flag**：ICMPv6 ルータ アドバタイズメント メッセージで、ホストがアドレス以外の関連情報を取得するためにステートフル自動構成を使用することを示します。 • **prefix**：ルータ アドバタイズメント メッセージで IPv6 プレフィックスをアドバタイズします。

	コマンドまたはアクション	目的
		• **ra-interval**：ICMPv6 ルータ アドバタイズメント メッセージの送信間の間隔を構成します。範囲は 4 ～ 1800 秒です。 • **ra-lifetime**：ICMPv6 ルータ アドバタイズメント メッセージで、デフォルト ルータのライフタイムをアドバタイズします。範囲は 0 ～ 9000 秒です。 • **reachable-time time**：ICMPv6 ルータ アドバタイズメント メッセージで、ノードが到達可能性確認を受信したあとにネイバーをアップしていると見なした時間をアドバタイズします。範囲は 0 ～ 9000 秒です。 • **redirects**：ICMPv6 リダイレクト メッセージの送信を有効にします。 （注） IPv6 リダイレクトを無効にする場合は、一部の IPv6 パケットが CPU にリークされる可能性があるため、IPv4 リダイレクトも無効にする必要があります。 • **retrans-timer**：time：ICMPv6 ルータ アドバタイズメント メッセージで、ネイバー送信要求メッセージ間の時間をアドバタイズします。範囲は 0 ～ 9000 秒です。 • **suppress-ra**：ICMPv6 ルータ アドバタイズメント メッセージの送信を無効にします。
Step 4	（任意） **show ip nd interface** 例： `switch(config-if)# show ip interface`	IPv6 ネイバー検出に構成されたインターフェイスを表示します。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

例

次に、IPv6 ネイバー探索到達可能時間を構成する例を示します。

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

次に、IPv6 インターフェイスを表示する例を示します。

```
switch# configure terminal
switch(config)# 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 unreachables: false
```

オプションの IPv6 ネイバー探索

次のオプションの IPv6 ネイバー探索コマンドを使用できます。

表 16 :

コマンド	目的
ipv6 nd hop-limit	ルータアドバタイズメントおよびルータから発信されるすべての IPv6 パケットで使用されるホップの最大数を構成します。
ipv6 nd managed-config-flag	「managed address configuration flag」フラグは、IPv6 ルータアドバタイズメントで設定されません。
ipv6 nd mtu	各インターフェイスにおいて送信される IPv6 パケットの最大伝送単位（MTU）サイズを設定します。
ipv6 nd ns-interval	インターフェイスで IPv6 ネイバー再送信要求メッセージが送信される時間間隔を設定します。

コマンド	目的
ipv6 nd other-config-flag	IPv6 ルータ アドバタイズメントに「other stateful configuration」フラグを構成します。
ipv6 nd ra-interval	インターフェイスで IPv6 ルータ アドバタイズメント（RA）メッセージが送信される時間間隔を構成します。
ipv6 nd ra-lifetime	インターフェイス上の IPv6 ルータ アドバタイズメントに含まれるルータのライフタイム値を構成します。
ipv6 nd reachable-time	到達可能性確認イベントがいくつか発生した後、リモート IPv6 ノードが到達可能と見なされる時間を構成します。
ipv6 nd redirects	ICMPv6 リダイレクト メッセージの送信を有効にします。
ipv6 nd retrans-timer	ルータ アドバタイズメントのネイバー送信要求メッセージ間のアドバタイズ時間を構成します。
ipv6 nd suppress-ra	LAN インターフェイス上で IPv6 ルータ アドバタイズメントの送信を抑制します。

IPv6 パケット検証の構成

Cisco NX-OS は、IPv6 パケットの検証をチェックする侵入検知システム（IDS）をサポートしています。これらの IDS チェックは有効または無効にできます。

IDS チェックを有効にするには、グローバル構成モードで次のコマンドを使用します。

表 *17*：

hardware ip verify address {**destination zero** | **identical** | **reserved** | **source multicast** }	IPv6 アドレスに対して次の IDS チェックを実行します。 • **destination zero**：宛先 IP アドレスが :: である場合は IPv6 パケットをドロップします。 • **identical**：送信元 IPv6 アドレスが宛先 IPv6 アドレスと同じである場合は IPv6 パケットをドロップします。 • **reserved**：IPv6 アドレスが ::1 である場合は、IPv6 パケットをドロップします。 • **source multicast**：送信元 IPv6 アドレスが FF00::/8 の範囲内（マルチキャスト）である場合は IPv6 パケットをドロップします。
hardware ipv6 verify length {**consistent** | **maximum** { **max-frag** | **max-tcp** | **udp** }}	IPv6 アドレスに対して次の IDS チェックを実行します。 • **consistent**：イーサネット フレーム サイズが、IPv6 パケット長にイーサネット ヘッダーを加えた値以上の場合には、IPv6 パケットをドロップします。 • **maximum max-frag**：計算式（IPv6 ペイロード長 - IPv6 拡張ヘッダー バイト数）+（フラグメント オフセット * 8）の値が 65536 より大きい場合には、IPv6 パケットをドロップします。 • **maximum max-tcp**：TCP 長が IP ペイロード長より長い場合は、IP パケットをドロップします。 • **maximum udp**：IPv6 ペイロード長が UDP パケット長を下回る場合には、IPv6 パケットをドロップします。
hardware ipv6 verify tcp tiny-frag	IPv6 フラグメント オフセットが 1 の場合、または IPv6 フラグメント オフセットが 0 で IP ペイロード長が 16 未満の場合、TCP パケットをドロップします。
hardware ipv6 verify version	EtherType が 6（IPv6）に設定されていない場合、IPv6 パケットをドロップします。

IPv6 パケット検証の構成を表示するには、show hardware forwarding ip verify コマンドを使用します。

IPv6 ステートレス自動構成の定義

手順の概要

1. **configure terminal**
2. **interface management** *number*
3. **ipv6 address autoconfig**
4. **ipv6 address autoconfig default**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例： `Device# configure terminal`	グローバルコンフィギュレーションモードを開始します。
Step 2	**interface management** *number* 例： `switch(config)# interface mgmt0`	インターフェイスのタイプと番号を指定し、デバイスをインターフェイスコンフィギュレーションモードにします。
Step 3	**ipv6 address autoconfig** 例： `switch(config-if)# ipv6 address autoconfig`	管理インターフェイスでステートレス自動構成を使用して、IPv6 アドレスの自動構成を有効にします。
Step 4	**ipv6 address autoconfig default** 例： `switch(config-if)# ipv6 address autoconfig default`	管理インターフェイスでステートレス自動構成を使用して IPv6 アドレスの自動構成を有効にし、ルータアドバタイズメントで受信したリンクローカル アドレスのネクストホップを持つデフォルト ルートを追加します。

例

次に、show ipv6 interface コマンドを使用して、管理インターフェイスで IPv6 アドレスが構成されていることを表示および確認する例を示します。[情報（Information）] には、SLAAC で生成されたアドレスを含む、インターフェイスに構成されているすべての IPV6 アドレスが表示されます。また、ステートレス アドレスの自動構成がインターフェイスで有効になっているかどうかも示します。

```
Device# show ipv6 interface mgmt 0

IPv6 Interface Status for VRF "management"(2)
mgmt0, Interface status: protocol-up/link-up/admin-up, iod: 2
IPv6 address:
1955::2f6:63ff:fe8b:c9f8/64 [VALID]
IPv6 subnet: 1955::/64
IPv6 link-local address: fe80::2f6:63ff:fe8b:c9f8 (default) [VALID]
…..
Stateless autoconfig configured on the interface
```

This example shows how to use the show ipv6 route vrf management command to display the IPv6 routing table for VRF management:

```
Device# 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]
0::/0, ubest/mbest: 1/0
*via fe80::2f6:63ff:fe8b:c9ff, mgmt0, [2/0], 00:02:00, icmpv6
1955::/64, ubest/mbest: 1/0, attached
*via 1955::2f6:63ff:fe8b:c9f8, mgmt0, [0/0], 15:59:22, direct,
1955::2f6:63ff:fe8b:c9f8/128, ubest/mbest: 1/0, attached
*via 1955::2f6:63ff:fe8b:c9f8, mgmt0, [0/0], 15:59:22, local
```

This example shows how to use the show ipv6 nd int mgmt command to display the ICMPv6 ND interfaces for VRF management:

```
Device# show ipv6 nd int mgmt 0

ICMPv6 ND Interfaces for VRF "management"
mgmt0, Interface status: protocol-up/link-up/admin-up
IPv6 address:
1955::2f6:63ff:fe8b:c9f8/64 [VALID]
IPv6 link-local address: fe80::2f6:63ff:fe8b:c9f8 [VALID]
…….
Subnets configured via SLAAC and their states:
Prefix 1955::/64[PREFERRED] Preferred lifetime left: 6d23h Valid lifetime lef
t: 4w1d
```

LPMヘビールーティングモードの設定（CiscoNexus9200および9300-EX プラットフォーム スイッチおよび 9732C-EX ライン カードのみ）

Cisco NX-OS リリース 7.0(3)I4(4) 以降では、極めて多くの LPM ルート エントリをサポートするために LPM のヘビー ルーティング モードを設定できます。このルーティング モードをサポートするのは、Cisco Nexus 9200 および 9300-EX シリーズのスイッチと、9732C-EX ライン カードを搭載した Cisco Nexus 9508 スイッチだけです。

（注） この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

（注） LPMヘビー ルーティング モードのスケール数については、『Cisco Nexus 9000 Series NX-OS Verified Scalability Guide』を参照してください。

手順の概要

1. **configure terminal**
2. **[no] system routing template-lpm-heavy**
3. （任意） **show system routing mode**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**[no] system routing template-lpm-heavy** 例： `switch(config)# system routing template-lpm-heavy`	デバイスをLPMヘビー ルーティング モードにして、より大きな LPM スケールをサポートします。

Step 3	（任意） **show system routing mode**
	例：
	`switch(config)# show system routing mode`
	`Configured System Routing Mode: LPM Heavy`
	`Applied System Routing Mode: LPM Heavy`
		LPM ルーティング モードを表示します。

Step 4	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。
Step 5	**reload** 例： `switch(config)# reload`	デバイス全体をリブートします。

LPM インターネット ピアリング ルーティング モードの設定（Cisco Nexus 9500-R プラットフォーム スイッチ、Cisco Nexus 9300-EX プラッ

トフォーム スイッチ、および Cisco Nexus 9000 シリーズ スイッチと 9700-EX ライン カードのみ）

Cisco NX-OS リリース7.0(3)I6(1) 以降では、IPv4 および IPv6 LPM インターネット ルート エントリをサポートするために LPM インターネット ピアリング ルーティング モードを設定できます。このモードは、IPv4 プレフィックス（/32 までのプレフィックス長）および IPv6 プレフィックス（/83 までのプレフィックス長）のダイナミック トライ（ツリー ビット ルックアップ）をサポートします。Cisco Nexus 9300-EX プラットフォーム スイッチおよび 9700-EXライン カードを搭載した Cisco Nexus 9500 プラットフォーム スイッチのみこのルーティング モードをサポートしています。

Cisco NX-OS リリース 9.3(1) 以降、Cisco Nexus 9500-R プラットフォーム スイッチはこのルーティング モードをサポートします。

（注） この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

（注） LPM インターネット ピアリング ルーティング モードのスケール数については、『Cisco Nexus 9000 Series NX-OS Verified Scalability Guide』を参照してください。LPM インターネット ピアリング モードの Cisco Nexus 9500-R プラットフォーム スイッチは、インターネット ピアリング プレフィックスを使用する場合にのみ、予測どおりにスケールアウトします。Cisco Nexus 9500-R プラットフォーム スイッチが他のプレフィックス パターンを使用している場合は、文書化されたスケーラビリティの数値を達成できない可能性があります。

手順の概要

1. **configure terminal**
2. [**no**] **system routing template-internet-peering**
3. （任意） **show system routing mode**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

	コマンドまたはアクション	目的
Step 2	[**no**] **system routing template-internet-peering** 例： `switch(config)# system routing template-internet-peering`	デバイスを LPM インターネット ピア ルーティング モードにして、IPv4 および IPv6 LPMインターネット ルート エントリをサポートします。

Step 3	（任意） **show system routing mode**
	例：
	`switch(config)# show system routing mode`
	`Configured System Routing Mode: Internet Peering`
	`Applied System Routing Mode: Internet Peering`
		LPM ルーティング モードを表示します。

Step 4	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。
Step 5	**reload** 例： `switch(config)# reload`	デバイス全体をリブートします。

LPM インターネット ピアリング ルーティング モードの追加設定

大規模ルーティング環境で LPM インターネット ピアリング ルーティング モードで Cisco Nexus スイッチを導入する場合、またはネクストホップ数が増加するルートの場合は、VDC リソーステンプレートで IPv4 のメモリ制限を増やす必要があります。

手順の概要

1. **configure terminal**
2. （任意） **show routing ipv4 memory estimate routes routes next-hops hops**
3. **vdc switch id id**
4. **limit-resource u4route-mem minimum min-limit maximum max-limit**
5. **exit**
6. **copy running-config startup-config**
7. **reload**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例：	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	`switch# configure terminal` `switch(config)#`	

Step 2	（任意） **show routing ipv4 memory estimate routes routes next-hops hops**
	例：
	`switch(config)# show routing ipv4 memory estimate routes 262144 next-hops 32`
	`Shared memory estimates:`
	`Current max 512 MB; 78438 routes with 64 nhs`
	`in-use 2 MB; 2642 routes with 1 nhs (average)`
	`Configured max 512 MB; 78438 routes with 64 nhs`
	`Estimate memory with fixed overhead: 1007 MB; 262144 routes with 32 nhs`
	`Estimate with variable overhead included:`
	`- With MVPN enabled VRF: 1136 MB`
	`- With OSPF route (PE-CE protocol): 1375 MB`
	`- With EIGRP route (PE-CE protocol): 1651 M`
		共有メモリの見積もりを表示して、ルートのメモリ要件を判断します。

Step 3	**vdc switch id id**
	例：
	`switch(config)# vdc switch id 1`
	`switch(config-vdc)#`
		VDC スイッチ ID を指定します。

Step 4	**limit-resource u4route-mem minimum min-limit maximum max-limit** 例： `switch(config-vdc)# limit-resource u4route-mem minimum 1024 maximum 1024`	IPv4 メモリの制限をメガバイト単位で指定します。 （注） Cisco Nexus リリース 10.2(2)F 以降、このコマンドは 32 ビット バージョンのソフトウェアにのみ適用されます。

Step 5	**exit**
	例：
	`switch(config-vdc)# exit`
	`switch(config)#`
		VDC 設定モードを終了します。

Step 6	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。
Step 7	**reload** 例： `switch(config)# reload`	デバイス全体をリブートします。

LPM デュアルホスト ルーティング モードの設定（Cisco Nexus 9200 および 9300-EX プラットフォーム スイッチ）

より多くの LPM ルート エントリをサポートするために、LPM ヘビー ルーティング モードを設定できます。このルーティング モードをサポートするのは、Cisco Nexus 9200 および 9300-EX プラットフォーム スイッチと、9732C-EX ライン カードを搭載した Cisco Nexus 9508 スイッチだけです。

（注） この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

（注） LPM ヘビー ルーティング モードのスケール数については、『Cisco Nexus 9000 Series NX-OS Verified Scalability Guide』を参照してください。

手順の概要

1. **configure terminal**
2. **[no] system routing template-lpm-heavy**
3. （任意） **show system routing mode**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**[no] system routing template-lpm-heavy** 例： `switch(config)# system routing template-lpm-heavy`	デバイスを LPM ヘビー ルーティング モードにして、より大きな LPM スケールをサポートします。

Step 3	（任意） **show system routing mode**
	例：
	`switch(config)# show system routing mode`
	`Configured System Routing Mode: LPM Heavy`
	`Applied System Routing Mode: LPM Heavy`
		LPM ルーティング モードを表示します。

	コマンドまたはアクション	目的
Step 4	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。
Step 5	**reload** 例： `switch(config)# reload`	デバイス全体をリブートします。

IPv6 リダイレクト Syslog の構成

IPv6 リダイレクト Syslog を有効/無効にするか、ログ間隔を変更するには、次の CLI を使用します。

（注） デフォルトでは、syslog のリダイレクトが有効になっています。

手順の概要

1. **configure terminal**
2. **ipv6 redirect syslog** [<*value*>]
3. （任意） **no ipv6 redirect syslog**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します

Step 2	**ipv6 redirect syslog** [<*value*>]
	例：
	`switch(config)# ip redirect syslog 60`
	`switch(config)#`
		過剰な IPv6 リダイレクトメッセージの syslog を構成します。
		• **ipv6 redirect syslog**: IPv6 リダイレクト メッセージの syslog を有効にします。
		• *value*: ログ間隔を設定します。範囲は最小 30 秒から最大 1800 秒です。デフォルトインターバルは 60 秒です。

	コマンドまたはアクション	目的
Step 3	（任意） **no ipv6 redirect syslog** 例： `switch(config)# no ipv6 redirect syslog`	過剰な IPv6 リダイレクト メッセージの syslog を無効にします。

IPv6 設定の確認

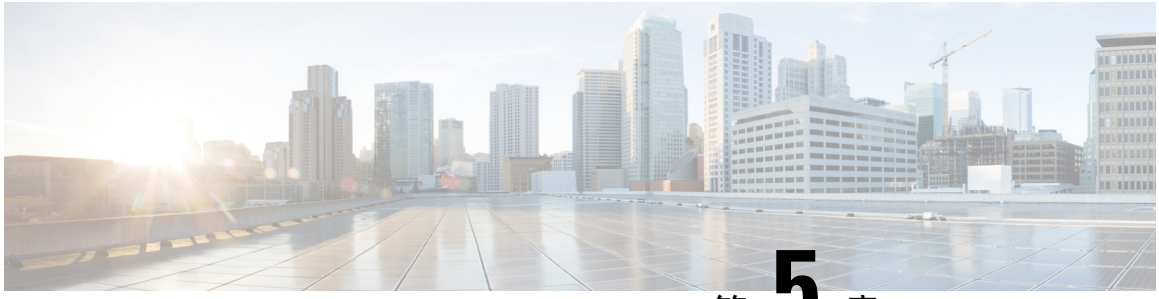
IPv6 設定を表示するには、次のいずれかの作業を行います。

コマンド	目的
show hardware forwarding ip verify	IPv4 および IPv6 パケット検証の構成を表示します。
show ipv6 interface	IPv6-related インターフェイスの情報を表示します。
show ipv6 adjacency	隣接関係テーブルを表示します。
show system routing mode	LPM ルーティング モードを表示します。
show ipv6 icmp	ICMPv6 情報を表示します。
show ipv6 nd	IPv6 ネイバー探索インターフェイス情報を表示します。
show ipv6 neighbor	IPv6 ネイバー エントリを表示します。
show ipv6 nd addr-registry	コンピューティング ノードの IPv6 アドレス レジストリ エントリを表示します。

IPv6 の設定例

次の例は IPv6 の設定方法を示しています。

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# ipv6 address 2001:db8::/64 eui64
switch(config-if)# ipv6 nd reachable-time 10
```



第 5 章

DNS の設定

この章では、Cisco NX-OS デバイスのドメイン ネーム サーバ（DNS）クライアントを設定する手順について説明します。

この章は、次の項で構成されています。

- DNS クライアントについて （105 ページ）
- 高可用性 （106 ページ）
- 仮想化のサポート （106 ページ）
- DNS クライアントの前提条件 （107 ページ）
- DNS クライアントに関する注意事項と制約事項 （107 ページ）
- DNS クライアントのデフォルト設定 （107 ページ）
- DNS クライアントの設定 （107 ページ）

DNS クライアントについて

DNS クライアントの概要

自分で名前の割り当てを管理していないネットワーク内のデバイスとの接続を、ネットワーク デバイスが必要とする場合は、DNS を使用して、ネットワーク間でデバイスを特定する一意のデバイス名を割り当てることができます。DNS は、階層方式を使用して、ネットワーク ノードのホスト名を確立します。これにより、クライアントサーバ方式によるネットワークのセグメントのローカル制御が可能となります。DNS システムは、デバイスのホスト名をその関連する IP アドレスに変換することで、ネットワーク デバイスを検出できます。

インターネット上のドメインは、組織のタイプや場所に基づく一般的なネットワークのグループを表す命名階層ツリーの一部です。ドメイン名は、ピリオド（.）を区切り文字として使用して構成されています。たとえば、シスコは、インターネットでは *com* ドメインで表される営利団体であるため、そのドメイン名は *cisco.com* です。このドメイン内の特定のホスト名、たとえばファイル転送プロトコル（FTP）システムは *ftp.cisco.com* で識別されます。

ネーム サーバ

ネームサーバはドメイン名の動向を把握し、自身が完全な情報を持っているドメインツリーの部分を認識しています。ネームサーバは、ドメインツリーの他の部分の情報を格納している場合もあります。Cisco NX-OS 内の IP アドレスにドメイン名をマッピングするには、ホスト名を示し、ネーム サーバを指定して、DNS サービスをイネーブルにする必要があります。

Cisco NX-OS では、スタティックに IP アドレスをドメイン名にマッピングできます。また、1 つ以上のドメイン ネーム サーバを使用してホスト名の IP アドレスを見つけるよう、Cisco NX-OS を設定することもできます。

DNS の動作

ネーム サーバは、次に示すように、特定のゾーン内でローカルに定義されるホストの DNS サーバに対してクライアントが発行したクエリーを処理します。

- 権限ネーム サーバは、その権限ゾーン内のドメイン名を求める DNS ユーザ照会に、自身のホスト テーブル内にキャッシュされた永久的なエントリを使用して応答します。照会で求められているのが、自身の権限ゾーン内であるが、設定情報が登録されていないドメイン名の場合、権限ネーム サーバはその情報が存在しないと応答します。
- 権限ネームサーバとして設定されていないネームサーバは、以前に受信した照会への返信からキャッシュした情報を使用して、DNSユーザ照会に応答します。ゾーンの権限ネームサーバとして設定されたルータがない場合は、ローカルに定義されたホストを求めるDNSサーバへの照会には、正規の応答は送信されません。

ネームサーバは、特定のドメインに設定された転送パラメータおよびルックアップパラメータに従って、DNS 照会に応答します（着信 DNS 照会を転送するか、内部的に生成された DNS 照会を解決します）。

高可用性

Cisco NX-OS は、DNS クライアントのステートレス再起動をサポートしています。リブートまたはスーパーバイザ スイッチオーバーの後に、Cisco NX-OS は実行コンフィギュレーションを適用します。

仮想化のサポート

Cisco NX-OS は、同じシステム上で動作する、DNS クライアントの複数インスタンスをサポートしています。DNSクライアントを設定できます。任意で、各仮想ルーティングおよび転送（VRF）インスタンスで、異なる DNS クライアント設定を使用できます。

DNS クライアントの前提条件

DNS クライアントには次の前提条件があります。

- ネットワーク上に DNS ネーム サーバが必要です。

DNS クライアントに関する注意事項と制約事項

DNS クライアントの設定時の注意事項および制約事項は、次のとおりです。

- DNS クライアントは特定の VRF に設定します。VRF を指定しない場合、Cisco NX-OS はデフォルトの VRF を使用します。
- Cisco NX-OS リリース 7.0(3)I5(1) 以降、DNS は IPv6 アドレスをサポートします。
- **ip name-server** CLI の **source-interface** 機能は、すべての DME 対応プラットフォームでサポートされています。NX-OS リリース 10.4(3)F より前では、DME は C92348GC-X でサポートされていません。したがって、**source-interface** オプションはこのプラットフォームではサポートされていません。ただし、**source-interface** オプションは、NX-OS リリース 10.4(3)F 以降の C92348GC-X でサポートされています。

DNS クライアントのデフォルト設定

下記の表は、DNS クライアント パラメータのデフォルト設定の一覧です。

デフォルトの DNS クライアント パラメータ

パラメータ	デフォルト
DNS クライアント	有効（Enabled）

DNS クライアントの設定

DNS クライアントの設定

ネットワーク上の DNS サーバを使用するよう、DNS クライアントを設定できます。

始める前に

ネットワーク上にドメイン ネーム サーバがあることを確認します。

手順の概要

1. **configure terminal**
2. **ip host** *name address1* [*address2... address6*]
3. （任意） **ip domain-name** *name* [**use-vrf** *vrf-name*]
4. （任意） **ip domain-list** *name* [**use-vrf** *vrf-name*]
5. （任意） **ip name-server** *address1* [*address2... address6*] **[use-vrf** *vrf-name*]
6. （任意） **ip domain-lookup**
7. （任意） **show hosts**
8. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**ip host** *name address1* [*address2... address6*] 例： `switch(config)# ip host cisco-rtp 192.0.2.1`	ホスト名キャッシュに、6 つまでのスタティック ホスト名/アドレスマッピングを定義します。使用可能なアドレスは、IPv4 アドレスまたは IPv6 アドレスです。
Step 3	（任意） **ip domain-name** *name* [**use-vrf** *vrf-name*] 例： `switch(config)# ip domain-name myserver.com`	Cisco NX-OS で使用するデフォルトのドメイン名を定義し、不完全なホスト名のドメインを補完します。このドメイン名を設定した VRF でこのドメイン ネームを解決できない場合は、任意で、Cisco NX-OS がこのドメイン ネームを解決するために使用する VRF を定義することもできます。 Cisco NX-OS は、ドメイン名ルックアップを開始する前に、完全なドメイン名を含まないあらゆるホスト名にデフォルト ドメイン名を付加します。
Step 4	（任意） **ip domain-list** *name* [**use-vrf** *vrf-name*] 例： `switch(config)# ip domain-list mycompany.com`	Cisco NX-OS が非修飾ホスト名を完成させるために使用できる追加のドメイン名を定義します。このドメイン名を設定した VRF でこのドメイン ネームを解決できない場合は、任意で、Cisco NX-OS がこのドメイン ネームを解決するために使用する VRF を定義することもできます。 Cisco NX-OS はドメイン名ルックアップを開始する前に、ドメイン リスト内の各エントリに基づいて、

	コマンドまたはアクション	目的
		完全なドメイン名を含んでいないすべてのホスト名にこのドメイン名を付加します。Cisco NX-OS は、一致するものが見つかるまで、ドメイン リストの各エントリにこのプロセスを実行します。
Step 5	（任意） **ip name-server** *address1* [*address2... address6*] **[use-vrf** *vrf-name*] 例： `switch(config)# ip name-server 192.0.2.22`	最大 6 台のネーム サーバを定義します。使用可能なアドレスは、IPv4 アドレスまたは IPv6 アドレスです。 このネーム サーバを設定した VRF でこのネーム サーバに到達できない場合は、任意で、Cisco NX-OS がこのネーム サーバに到達するために使用する VRF を定義することもできます。 （注） 複数の DNS サーバは、応答しないサーバの場合に使用します。 リスト内の最初の DNS サーバが拒否で DNS クエリに応答した場合、残りの DNS サーバは照会されません。最初のサーバが応答しない場合、リスト内の次の DNS サーバが照会されます。
Step 6	（任意） **ip domain-lookup** 例： `switch(config)# ip domain-lookup`	DNS ベースのアドレス変換をイネーブルにします。この機能は、デフォルトでイネーブルにされています。
Step 7	（任意） **show hosts** 例： `switch(config)# show hosts`	DNS に関する情報を表示します。
Step 8	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、デフォルト ドメイン名を設定し、DNS ルックアップをイネーブルにする例を示します。

```
switch# configure terminal
switch(config)# ip domain-name cisco.com
switch(config)# ip name-server 192.0.2.1 use-vrf management
switch(config)# ip domain-lookup
switch(config)# copy running-config startup-config
```

仮想化の設定

VRF 内に DNS クライアントを設定できます。VRF コンフィギュレーション モードを使用しない場合は、ご使用の DNS クライアント設定がデフォルト VRF に適用されます。

または、DNS クライアントを設定した VRF 以外の、指定した VRF をバックアップ VRF として使用するよう、DNS クライアントを設定することもできます。たとえば、DNS クライアントを赤の VRF で設定していても、赤の VRF で DNS サーバに到達できない場合は、青の VRF を使用して DNS サーバと通信できます。

始める前に

ネットワーク上にドメイン ネーム サーバがあることを確認します。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf-name*
3. （任意） **ip domain-name** *name* [**use-vrf** *vrf-name*]
4. （任意） **ip domain-list** *name* [**use-vrf** *vrf-name*]
5. （任意） **ip name-server** *address1* [*address2... address6*] [**use-vrf** *vrf-name*]
6. （任意） **show hosts**
7. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context Red`
	`switch(config-vrf)#`
		VRF を作成し、VRF設定モードを開始します。

Step 3	（任意） **ip domain-name** *name* [**use-vrf** *vrf-name*] 例： `switch(config-vrf)# ip domain-name myserver.com`	Cisco NX-OS で使用するデフォルトのドメイン名サーバを定義し、不完全なホスト名のドメインを補完します。このドメイン名を設定した VRF でこのドメイン ネーム サーバを解決できない場合は、任意で、Cisco NX-OS がこのドメイン ネーム サーバを解決するために使用する VRF を定義することもできます。

	コマンドまたはアクション	目的
		Cisco NX-OS は、ドメイン名ルックアップを開始する前に、完全なドメイン名を含まないあらゆるホスト名にデフォルト ドメイン名を付加します。
Step 4	（任意） **ip domain-list** *name* [**use-vrf** *vrf-name*] 例： `switch(config-vrf)# ip domain-list mycompany.com`	Cisco NX-OS が非修飾ホスト名を完成させるために使用できる追加のドメイン名サーバを定義します。このドメイン名を設定した VRF でこのドメイン ネーム サーバを解決できない場合は、任意で、Cisco NX-OS がこのドメイン ネーム サーバを解決するために使用する VRF を定義することもできます。 Cisco NX-OS はドメイン名ルックアップを開始する前に、ドメイン リスト内の各エントリに基づいて、完全なドメイン名を含んでいないすべてのホスト名にこのドメイン名を付加します。Cisco NX-OS は、一致するものが見つかるまで、ドメイン リストの各エントリにこのプロセスを実行します。
Step 5	（任意） **ip name-server** *address1* [*address2... address6*] [**use-vrf** *vrf-name*] 例： `switch(config-vrf)# ip name-server 192.0.2.22`	最大 6 台のネーム サーバを定義します。使用可能なアドレスは、IPv4 アドレスまたは IPv6 アドレスです。 このネーム サーバを設定した VRF でこのネーム サーバに到達できない場合は、任意で、Cisco NX-OS がこのネーム サーバに到達するために使用する VRF を定義することもできます。 （注） 複数の DNS サーバは、応答しないサーバの場合に使用します。 リスト内の最初の DNS サーバが拒否で DNS クエリに応答した場合、残りの DNS サーバは照会されません。最初のサーバが応答しない場合、リスト内の次の DNS サーバが照会されます。
Step 6	（任意） **show hosts** 例： `switch(config-vrf)# show hosts`	DNS に関する情報を表示します。
Step 7	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、デフォルト ドメインを設定し、VRF 内の DNS ルックアップを有効にする例を示します。

```
switch# configure terminal
switch(config)# vrf context Red
switch(config-vrf)# ip domain-name cisco.com
switch(config-vrf)# ip name-server 192.0.2.1 use-vrf management
switch(config-vrf)# copy running-config startup-config
```

DNS クライアントの設定の確認

DNS クライアントの設定を表示するには、次のいずれかの作業を行います。

コマンド	目的
show hosts	DNS に関する情報を表示します。

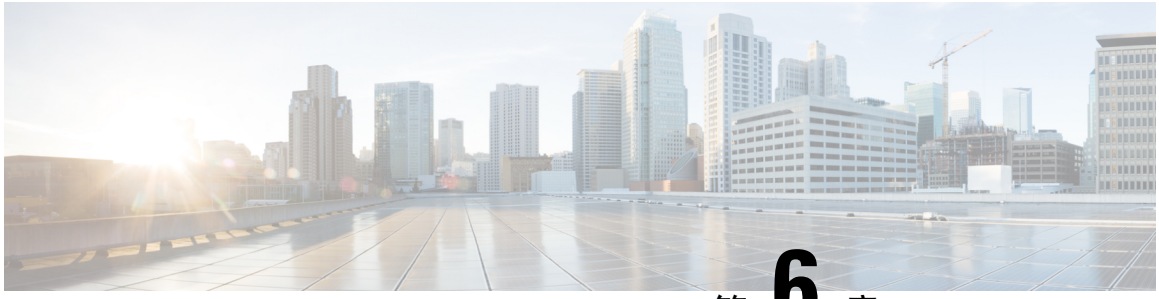
DNS クライアントの設定例

次の例は、複数の代替ドメイン名があるドメイン リストの設定方法を示しています。

```
ip domain-list csi.com
ip domain-list telecomprog.edu
ip domain-list merit.edu
```

次に、ホスト名とアドレス間のマッピングプロセスを設定し、IP DNS ベースの変換を指定する例を示します。例では、ネーム サーバとデフォルトのドメイン名のアドレスを設定します。

```
ip domain-lookup
ip name-server 192.168.1.111 192.168.1.2
ip domain-name cisco.com
```



第 6 章

OSPFv2 の設定

この章では、Cisco NX-OS デバイスで IPv4 ネットワーク用の Open Shortest Path First version 2（OSPFv2）を設定する方法について説明します。

この章は、次の項で構成されています。

- OSPFv2 について （113 ページ）
- OSPFv2およびユニキャストRIB （114 ページ）
- 認証 （114 ページ）
- 高度な機能 （116 ページ）
- OSPFv2 の前提条件 （121 ページ）
- OSPFv2 の注意事項および制約事項 （121 ページ）
- OSPFv2のデフォルト設定 （123 ページ）
- 基本的な OSPFv2 の設定 （124 ページ）
- 高度なOSPFv2の設定 （136 ページ）
- OSPFv2 設定の確認 （165 ページ）
- OSPFv2 のモニタリング （166 ページ）
- OSPFv2 の設定例 （166 ページ）
- その他の参考資料 （167 ページ）

OSPFv2 について

OSPFv2 は、IPv4 ネットワーク用 IETF リンクステート プロトコルです（「リンクステート プロトコル」の項を参照）。OSPFv2 ルータは、hello パケットと呼ばれる特別なメッセージを各 OSPF 対応インターフェイスに送信して、ほかの OSPFv2 隣接ルータを探索します。ネイバー ルータが発見されると、この 2 台のルータは hello パケットの情報を比較して、両者の設定に互換性のあるかどうかを判定します。これらの隣接ルータは隣接を確立しようとします。つまり、両者のリンクステート データベースを同期させて、確実に同じ OSPFv2 ルーティング情報を持つようにします。隣接ルータは、各リンクの稼働状態に関する情報、リンクのコスト、およびその他のあらゆるネイバー情報を含むリンクステート アドバタイズメント（LSA）を共有します。これらのルータはその後、受信した LSA をすべての OSPF 対応インターフェイスにフラッディングします。これにより、すべての OSPFv2 ルータのリンクステート データベースが最終的に同じになります。

すべての OSPFv2 ルータのリンクステート データベースが同じになると、ネットワークは収束します（「コンバージェンス」を参照）。その後、各ルータは、ダイクストラの最短パス優先（SPF）アルゴリズムを使用して、自身のルート テーブルを構築します。

OSPFv2 ネットワークは、複数のエリアに分割できます。ルータは、ほとんどの LSA を 1 つのエリア内だけに送信するため、OSPF 対応ルータの CPU とメモリの要件が緩やかになります。

OSPFv2 は IPv4 をサポートし、OSPFv3 は IPv6 をサポートしています。詳細については、OSPFv3 の設定 （169 ページ）を参照してください。

（注） Cisco NX-OS 上の OSPFv2 は、RFC 2328 をサポートしています。この RFC では、ルート サマリー コストの計算に、RFC1583 で使用する計算と互換性がない別の方法が導入されました。また RFC 2328 では、AS-external パスに対して異なる選択基準が導入されました。すべてのルータが同じ RFC をサポートしていることを確認することが重要です。RFC。RFC1583 にのみ準拠しているルータがネットワークに含まれる場合は、**rfc1583compatibility** コマンドを使用します。デフォルトでサポートされている OSPFv2 用の RFC 標準は、Cisco NX-OS と Cisco IOS とで異なる場合があります。値が同じになるように設定するには、調整が必要です。詳細については、「OSPF RFC 互換モードの例」の項を参照してください。

OSPFv2およびユニキャストRIB

OSPFv2 は、リンクステート データベースでダイクストラの SPF アルゴリズムを実行します。このアルゴリズムにより、パス上の各リンクのリンク コストの合計に基づいて、各宛先への最適なパスが選択されます。そして、選択された各宛先への最短パスが OSPFv2 ルート テーブルに入力されます。OSPFv2 ネットワークが収束すると、このルート テーブルはユニキャスト RIB にデータを提供します。OSPFv2 はユニキャスト RIB と通信し、次の動作を行います。

- ルートの追加または削除
- 他のプロトコルからのルートの再配布への対応
- 変更されていない OSPFv2 ルートの削除およびスタブ ルータ アドバタイズメントを行うためのコンバージェンス更新情報の提供（「OSPFv2 スタブ ルータ アドバタイズメント」セクションを参照）

さらに OSPFv2 は、変更済みダイクストラ アルゴリズムを実行して、集約および外部（タイプ 3、4、5、7）LSA の変更の高速再計算を行います。

認証

OSPFv2 メッセージに認証を設定して、ネットワークでの不正な、または無効なルーティング更新を防止できます。Cisco NX-OS は、次の 2 つの認証方式をサポートしています。

- 簡易パスワード認証

- MD5 認証ダイジェスト

OSPFv2 認証は、OSPFv2 エリアに対して、またはインターフェイスごとに設定できます。

簡易パスワード認証

簡易パスワード認証では、OSPFv2 メッセージの一部として送信された単純なクリア テキストのパスワードを使用します。受信 OSPFv2 ルータが OSPFv2 メッセージを有効なルート更新情報として受け入れるには、同じクリア テキスト パスワードで設定されている必要があります。パスワードがクリア テキストであるため、ネットワーク上のトラフィックをモニタできるあらゆるユーザがパスワードを入手できます。

暗号化認証

暗号化認証では、暗号化されたパスワードをOSPFv2認証に使用します。トランスミッタは、送信するパケットとキー文字列を使用してコードを計算し、そのコードとキーIDをパケットに挿入して、パケットを送信します。受信側は、受信したパケットとローカルに設定されたキーストリング（パケット内のキーIDに対応）を使用してコードをローカルに計算することにより、パケット内のコードを検証します。

メッセージダイジェスト 5（MD5）とハッシュベースのメッセージ認証コードセキュアハッシュアルゴリズム（HMAC-SHA）暗号化認証の両方がサポートされています。

MD5 認証

OSPFv2 メッセージを認証するには、MD5 認証を使用する必要があります。そのためには、ローカルルータとすべてのリモートOSPFv2ネイバーが共有するパスワードを設定します。Cisco NX-OS は各 OSPFv2 メッセージに対して、メッセージと暗号化されたパスワードに基づく MD5 一方向メッセージダイジェストを作成します。インターフェイスはこのダイジェストをOSPFv2メッセージとともに送信します。受信する OSPFv2 ネイバーは、同じ暗号化パスワードを使用して、このダイジェストを確認します。メッセージが変更されていない場合はダイジェストの計算が同一であるため、OSPFv2 メッセージは有効と見なされます。

MD5 認証には、ネットワークでのメッセージの再送を防ぐための、各 OSPFv2 メッセージのシーケンス番号が含まれます。

HMAC-SHA 認証

Cisco NX-OS リリース 7.0（3）I3（1）以降、OSPFv2 は RFC 5709 をサポートしており、MD5 よりも高いセキュリティを提供する HMAC-SHA アルゴリズムを使用できます。HMAC-SHA-1、HMAC-SHA-256、HMAC-SHA-384。および HMAC-SHA-512 アルゴリズムは、OSPFv2認証でサポートされます。

高度な機能

Cisco NX-OS は、ネットワークでの OSPFv2 の可用性やスケーラビリティを向上させる、高度な OSPFv3 機能をサポートしています。

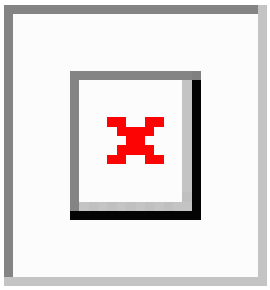
スタブ エリア

エリアをスタブ エリアにすると、エリアでフラッディングされる外部ルーティング情報の量を制限できます。スタブエリアとは、AS外部（タイプ5）LSA（リンクステートアドバタイズメント（174 ページ）の項を参照）が許可されないエリアです。これらの LSA は通常、外部ルーティング情報を伝播するためにローカル自律システム全体でフラッディングされます。スタブ エリアには、次の要件があります。

- スタブ エリア内のすべてのルータはスタブ ルータです。「スタブ ルーティング」の項を参照してください。
- スタブ エリアには ASBR ルータは存在しません。
- スタブ エリアには仮想リンクを設定できません。

次の図には、外部 AS に到達するためにエリア 0.0.0.10 内のすべてのルータが ABR を通過する必要のある OSPFv2 AS の例を示します。エリア 0.0.0.10 は、スタブ エリアとして設定できます。

図 *20*：スタブ エリア



スタブエリアは、外部自律システムへのバックボーンエリアを通過する必要のあるすべてのトラフィックにデフォルト ルートを使用します。IPv4 の場合のデフォルト ルートは 0.0.0.0 です。

Not So Stubby Area

Not-So-Stubby Area（NSSA）は、スタブ エリアに似ていますが、NSSA では、再配布を使用して NSSA 内で自律システム外部ルートをインポートできる点が異なります。NSSA ASBR はこれらのルートを再配布し、NSSA 外部（タイプ 7）LSA を生成して NSSA 全体でフラッディングします。または、NSSA を他のエリアに接続する ABR を設定することにより、この NSSA 外部 LSA を AS 外部（タイプ 5）LSA に変換することもできます。こうすると、ABR は、これらの AS 外部 LSA を OSPFv2 自律システム全体にフラッディングします。変換中は集約とフィルタリングがサポー

トされます。NSSA 外部 LSA に関する情報については、リンクステートアドバタイズメント（174 ページ）セクションを参照してください。

たとえば、OSPFv2 を使用する中央サイトを、異なるルーティング プロトコルを使用するリモート サイトに接続するときに NSSA を使用すると、管理作業を簡素化できます。リモート サイトへのルートはスタブ エリア内に再配布できないため、NSSA を使用する前に、企業サイトの境界ルータとリモート ルータの間の接続を OSPFv2 スタブ エリアとして実行できません。NSSA を使用すると、企業のルータとリモート ルータ間のエリアを NSSA として定義する（「NSSA の設定」を参照）ことで、OSPFv2 を拡張してリモート接続性をサポートできます。

バックボーン エリア 0 を NSSA にできません。

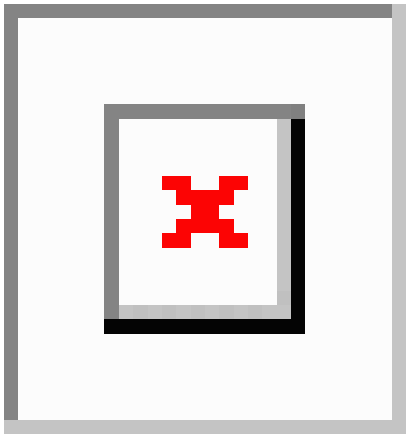
（注） Cisco NX-OS リリース 9.2(4) 以降、OSPF は RFC 3101 セクション 2.5(3) に準拠するようになりました。Not-so-Stubby Area に接続されたエリア境界ルータが P ビット クリアのデフォルト ルート LSA を受信した場合は、無視されます。OSPF は、これらの条件下で以前にデフォルト ルートを追加していました。

すでに RFC 非準拠の動作を使用するようにネットワークを設計しており、デフォルト ルートが NSSA ABR に追加されると想定している場合は、Cisco NX-OS リリース 9.2(4) 以降にアップグレードするときに動作が変更されます。

仮想リンク

仮想リンクを使用すると、物理的に直接接続できない場合に、OSPFv2 エリア ABR をバックボーン エリア ABR に接続できます。図には、エリア 3 をエリア 5 経由でバックボーン エリアに接続する仮想リンクを示します。

図 *21* : 仮想リンク



また、仮想リンクを使用して、分割エリアから一時的に回復できます。分割エリアは、エリア内のリンクがダウンしたために隔離された一部のエリアで、ここからはバックボーン エリアへの代表 ABR に到達できません。

ルートの再配布

OSPFv2は、ルート再配布を使用して、他のルーティングプロトコルからルートを学習できます。ルートの再配布の概要 （8 ページ） の項を参照してください。リンク コストをこれらの再配布されたルートに割り当てるか、またはデフォルトリンクコストを再配布されたすべてのに割り当てるよう、OSPFv2 を設定します。

ルート再配布では、ルート マップを使用して、再配布する外部ルートを管理します。再配布を指定したルートマップを設定して、どのルートがOSPFv2に渡されるかを制御する必要があります。ルート マップを使用すると、宛先、送信元プロトコル、ルート タイプ、ルート タグなどの属性に基づいて、ルートをフィルタリングできます。ルート マップを使用して、これらの外部ルートがローカルOSPFv2 自律システムでアドバタイズされる前にAS外部（タイプ 5）LSAおよびNSSA外部（タイプ 7）LSA のパラメータを変更できます。ルート マップの設定については、「Route Policy Manager の設定」を参照してください。

ルート集約

OSPFv2は、学習したすべてのルートを、すべてのOSPF対応ルータと共有するため、ルート集約を使用して、すべての OSPF 対応ルータにフラッディングされる一意のルートの数を削減した方がよい場合があります。ルート集約により、より具体的な複数のアドレスが、すべての具体的なアドレスを表す 1 つのアドレスに置き換えられるため、ルート テーブルが簡素化されます。たとえば、10.1.1.0/24、10.1.2.0/24、および10.1.3.0/24というアドレスを1つの集約アドレス10.1.0.0/16に置き換えることができます。

一般的には、エリア境界ルータ（ABR）の境界ごとに集約します。集約は 2 つのエリアの間でも設定できますが、バックボーンの方向に集約する方が適切です。こうすると、バックボーンがすべての集約アドレスを受信し、すでに集約されているそれらのアドレスを他のエリアに投入できるためです。集約には、次の 2 タイプがあります。

- エリア間ルート集約
- 外部ルート集約

エリア間ルート集約はABR上で設定し、自律システム内のエリア間のルートを集約します。集約の利点を生かすには、これらのアドレスを 1 つの範囲内にまとめることができるように、連続するネットワーク番号をエリア内で割り当てる必要があります。

外部ルート集約は、ルート再配布を使用して OSPFv2 に投入される外部ルートに特有のルート集約です。集約する外部の範囲が連続していることを確認する必要があります。異なる 2 台のルータからの重複範囲を集約すると、誤った宛先にパケットが送信される原因となる場合があります。外部ルート集約は、ルートを OSPF に再配布している ASBR で設定してください。

集約アドレスの設定時に Cisco NX-OS は、ルーティング ブラック ホールおよびルート ループを防ぐために、集約アドレスの廃棄ルートを自動的に設定します。

高可用性およびグレースフル リスタート

Cisco NX-OS は、マルチレベルの高可用性 アーキテクチャを提供します。OSPFv2 は、ステートフル リスタートをサポートしています。これは、ノンストップ ルーティング（NSR）とも呼ばれます。OSPFv2 で問題が発生した場合は、以前の実行時状態からの再起動を試みます。この場合、ネイバーはいずれのネイバー イベントも登録しません。最初の再起動が正常ではなく、別の問題が発生した場合、OSPFv2 はグレースフル リスタートを試みます。

グレースフル リスタート、つまり、Nonstop Forwarding（NSF）では、処理の再起動中も OSPFv2 がデータ転送パス上に存在し続けます。OSPFv2 はグレースフル リスタートを実行する必要がある場合、猶予 LSA と呼ばれるリンクローカル不透明（タイプ 9）LSA を送信します。この再起動中の OSPFv2 プラットフォームは NSF 対応と呼ばれます。

猶予 LSA には猶予期間が含まれます。猶予期間とは、ネイバー OSPFv2 インターフェイスが再起動中の OSPFv2 インターフェイスからの LSA を待つよう指定された時間です（通常、OSPFv2 は隣接関係を切断し、ダウン状態または再起動中の OSPFv2 インターフェイスからのすべての LSA を廃棄します）。参加するネイバーは、NSF ヘルパーと呼ばれ、再起動中の OSPFv2 インターフェイスから発信されたすべての LSA を、インターフェイスがまだ隣接しているかのように保持します。

再起動中の OSPFv2 インターフェイスが稼働を再開すると、ネイバーを再探索して隣接関係を確立し、LSA 更新情報の送信を再開します。この時点で、NSF ヘルパーは、グレースフル リスタートが完了したと認識します。

ステートフル リスタートは次のシナリオで使用されます。

- プロセスでの問題発生後の最初の回復試行
- **system switchover** を使用したユーザ開始スイッチオーバー command

グレースフル リスタートは次のシナリオで使用されます。

- プロセスでの問題発生後の 2 回目の回復試行（4 分以内）
- **restart ospf** を使用したプロセスの手動再起動 command
- アクティブ スーパーバイザの削除
- **reload module** *active-sup* コマンド

OSPFv2 スタブ ルータ アドバタイズメント

OSPFv2 スタブ ルータ アドバタイズメント機能を使用して、OSPFv2 インターフェイスをスタブ ルータとして機能するように設定できます。この機能は、ネットワークに新規ルータを機能制限付きで導入する場合や、過負荷になっているルータの負荷を制限する場合など、このルータ経由の OSPFv2 トラフィックを制限するときに使用します。また、この機能は、さまざまな管理上またはトラフィック エンジニアリング上の理由により使用する場合もあります。

OSPFv2 スタブ ルータ アドバタイズメントは、OSPFv2 ルータをネットワーク トポロジから削除しませんが、他の OSPFv2 ルータがこのルータを使用して、ネットワークの他の部分にトラフィッ

クをルーティングできないようにします。このルータを宛先とするトラフィック、またはこのルータに直接接続されたトラフィックだけが送信されます。

OSPFv2 スタブ ルータ アドバタイズメントは、すべてのスタブ リンク（ローカル ルータに直接接続された）を、ローカルOSPFv2インターフェイスのコストとしてマークします。すべてのリモート リンクは、最大のコスト（0xFFFF）としてマークされます。

複数の OSPFv2 インスタンス

Cisco NX-OS は、同じノード上で動作する、OSPFv2 プロトコルの複数インスタンスをサポートしています。同一インターフェイスには複数のインスタンスを設定できません。デフォルトでは、すべてのインスタンスが同じシステム ルータ ID を使用します。複数のインスタンスが同じ OSPFv2 自律システムにある場合は、各インスタンスのルータ ID を手動で設定する必要があります。サポートされる OSPFv2 インスタンスの数については、『Cisco Nexus 9000 Series NX-OS Verified Scalability Guide』を参照してください。

SPF 最適化

Cisco NX-OS は、次の方法で SPF アルゴリズムを最適化します。

- ネットワーク（タイプ 2）LSA、ネットワーク集約（タイプ 3）LSA、および AS 外部（タイプ 5）LSA 用の部分的 SPF：これらの LSA のいずれかが変更されると、Cisco NX-OS は、全体的な SPF 計算ではなく、高速部分計算を実行します。
- SPF タイマー：さまざまなタイマーを設定して、SPF 計算を制御できます。これらのタイマーには、後続の SPF 計算の幾何バックオフが含まれます。幾何バックオフにより、複数の SPF 計算による CPU 負荷が制限されます。

BFD

この機能では、双方向フォワーディング検出（BFD）をサポートします。BFD は、転送パスの障害を高速で検出することを目的にした検出プロトコルです。BFD は 2 台の隣接デバイス間のサブセカンド障害を検出し、BFD の負荷の一部を、サポートされるモジュール上のデータ プレーンに分散できるため、プロトコル hello メッセージよりも CPU を使いません。詳細については、『Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide』を参照してください。

OSPFv2 の仮想化のサポート

Cisco NX-OS は、OSPFv3 の複数のプロセス インスタンスをサポートします。各 OSPF インスタンスは、システム制限まで、複数の仮想ルーティングおよび転送（VRF）インスタンスをサポートできます。サポートされる OSPFv2 インスタンスの数については、『Cisco Nexus 9000 Series NX-OS Verified Scalability Guide』を参照してください。

OSPFv2 の前提条件

OSPFv2 には、次の前提条件があります。

- OSPF を設定するための、ルーティングの基礎に関する詳しい知識がある。
- スイッチにログインしている。
- リモート OSPFv2 ネイバーと通信可能な IPv4 用インターフェイスが 1 つ以上設定されている。
- OSPFv2 ネットワーク戦略と、ネットワークのプランニングが完成している。たとえば、複数のエリアが必要かどうかを決定します。
- OSPF 機能がイネーブルにされている（「OSPFv2 のイネーブル化 」の項を参照）。

OSPFv2 の注意事項および制約事項

OSPFv2 設定時の注意事項および制約事項は、次のとおりです。

- **reload** の OSPFv2 の **graceful-restart planned-only** コマンドは **graceful-restart** コマンドに変換されます。

  これは機能に影響を与えません。**graceful-restart planned-only** が設定にない場合、この問題はそのデバイスには適用されません。

  これは、Cisco NX-OS リリースが 9.3(2) で、CSCvs57583 がリリースに含まれていない場合に発生します。回避策は、**graceful-restart** コマンドを設定解除し、古いコマンドを再設定することです。

- プレフィックス リスト内の名前は、大文字と小文字が区別されません。一意の名前を使用することを推奨します。大文字と小文字を変更して同じ名前を使用しないでください。たとえば、CTCPrimaryNetworks と CtcPrimaryNetworks は 2 つの異なるエントリではありません。
- **no graceful-restart planned only** コマンドを入力すると、グレースフル リスタートは無効になります。
- Cisco NX-OS は、ユーザがエリアを 10 進表記で入力するか、ドット付き 10 進表記で入力するかに関係なく、ドット付き 10 進表記でエリアを表示します。
- すべての OSPFv2 ルータが、同じ RFC 互換モードで動作する必要があります。Cisco NX-OS の OSPFv2 は RFC 2328 に準拠しています。RFC 1583 にのみ対応しているルータがネットワークに含まれている場合は、ルータ設定モードで **rfc1583compatibility** コマンドを使用します。
- スケール シナリオでは、インターフェイスと OSPF プロセスのリンク ステート アドバタイズメントの数が大きい場合、OSPF MIB オブジェクトの SNMP エージェントのタイムアウト値が小さい SNMP ウォークは、タイムアウトになると予想されます。OSPF MIB オブジェクト

のポーリング中に問い合わせる SNMP エージェントのタイムアウトを確認する場合は、ポーリングする SNMP エージェントのタイムアウト値を増加してください。

- アドミニストレーティブ ディスタンス機能には、次のガイドラインと制限事項が適用されます。
  - OSPF ルートに複数の等コスト パスがある場合、アドミニストレーティブ ディスタンスを設定しても **match ip route-source** コマンドに対しては決定性を持ちません。
  - アドミニストレーティブディスタンスの設定は、**match route-type**、**match ip address prefix-list**、および **match ip route-source prefix-list** コマンドでのみサポートされます。別の match 文は無視されます。
  - OSPF ルートのアドミニストレーティブ ディスタンスを設定する場合、**match route-type**、**match ip address**、および **match ip route-source** コマンドの間に優先順位はありません。このように、Cisco NX-OS OSPF アドミニストレーティブ ディスタンスを設定するためのテーブル マップの動作は、Cisco IOS OSPF の場合と異なります。
  - 廃棄ルートには、アドミニストレーティブ ディスタンス 220 が常に割り当てられます。テーブル マップの設定は OSPF の廃棄ルートには適用されません。
- vPC 設定モードで **delay restore** *seconds* コマンドを設定する場合や、マルチシャーシ EtherChannel トランク（MCT）上の VLAN がスイッチ仮想インターフェイス（SVI）を使用して OSPFv2 または OSPFv3 によって通知される場合、これらの SVI は設定された時間の間、vPC セカンダリ ノード上で MAX_LINK_COST で通知されます。その結果、すべてのルートまたはホストのプログラミングは、トラフィックを引き込む前に（セカンダリ vPC ノードのピア リロードで）vPC の同期操作後に完了します。この動作により、ノースサウス トラフィックのパケット損失を最小にできます。
- N9K-X9636C-R および N9K-X9636Q-R ライン カードおよび N9K-C9508-FM-R ファブリックモジュールの場合、 **show run ospf** コマンドの出力には、一部の OSPF コマンドのデフォルト値が表示されることがあります。

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

- OSPF で **network** *ip address mask* コマンドを使用すると、エラー メッセージが表示され、**area** *area id* コマンドを使用してインターフェイスで OSPF を有効にするように求められます。
- OSPF のデフォルト タイマー（hello-interval:10 および dead-interval:40）を使用することをお勧めします。コンバージェンス時間を短縮するには、OSPF とともに BFD を使用できます。この組み合わせにより、1 秒未満のリンク/隣接フラップ検出と非常に短いコンバージェンス時間が実現します。

- OSPF サポートはアグレッシブタイマーですが、アグレッシブ イマーは調整をすぐにダウンさせ、CPU チャーンを引き起こすため、推奨されません。デフォルトのタイマーを使用し、BFD（双方向転送検出）を使用して 1 秒未満の障害検出を行うことを推奨します。
- Cisco NX-OS リリース 10.3(1)F 以降、OSPFv2 は Cisco Nexus 9808 スイッチでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、 OSPFv2 は Cisco Nexus 9804 スイッチでサポートされます。
- Cisco NX-OS リリース 10.3(3)F 以降、OSPFv2 は Cisco NX-OS スイッチの OSPFv2 ユーザー パスワードのタイプ 6 キーチェーン暗号化をサポートします。
- Cisco NX-OS リリース 10.4(1)F 以降、OSPFv2 は Cisco Nexus 9808 および 9804 スイッチを搭載した Cisco Nexus X98900CD-A および X9836DM-A ライン カードでサポートされます。

OSPFv2のデフォルト設定

次の表に、OSPFv2 パラメータのデフォルト設定値を示します。

表 *18 : OSPFv2* のデフォルト パラメータ

パラメータ	デフォルト
アドミニストレーティブ ディスタンス	110
hello 間隔	10 秒
デッド間隔	40 秒
廃棄ルート	イネーブル
グレースフル リスタートの猶予期間	60 秒
OSPFv2 機能	ディセーブル
スタブ ルータ アドバタイズメントの宣言期間	600 秒
リンク コスト計算の参照帯域幅	40 Gbps
LSA 最小到着時間	1000 ミリ秒
LSA グループ ペーシング	10 秒
SPF 計算初期遅延時間	200 ミリ秒
SPF の最小ホールド タイム	5000 ミリ秒
SPF 計算初期遅延時間	1000 ミリ秒

基本的な OSPFv2 の設定

OSPFv2 は、OSPFv2 ネットワークを設計した後に設定します。

OSPFv2の有効化

OSPFv2 を設定するには、その前に OSPFv2 機能を有効にする必要があります。

手順の概要

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

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**feature ospf** 例： `switch(config)# feature ospf` 例：	OSPFv2 機能を有効にします。
Step 3	（任意） **show feature** 例： `switch(config)# show feature`	有効および無効にされた機能を表示します。
Step 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

OSPFv2 機能をディセーブルにして、関連付けられている設定をすべて削除するには、グローバル設定モードで no feature ospf コマンドを使用します。

コマンド	目的
no feature ospf 例： `switch(config)# no feature ospf`	OSPFv2 機能を無効にして、関連付けられた設定をすべて削除します。

OSPFv2インスタンスの作成

OSPFv2 を設定する最初のステップは、OSPFv2 インスタンスを作成することです。作成した OSPFv2 インスタンスには、一意のインスタンス タグを割り当てます。インスタンス タグは任意の文字列です。

OSPFv2 インスタンス パラメータの詳細については、高度なOSPFv2の設定 （136 ページ）の項を参照してください。

始める前に

OSPF 機能をイネーブルにしてあることを確認します（「OSPFv2のイネーブル化」の項を参照）。

show ip ospf *instance-tag* コマンドを使用して、インスタンス タグが使用されていないことを確認します。

OSPFv2 がルータ ID（設定済みのループバック アドレスなど）を入手可能であるか、またはルータ ID オプションを設定する必要があります。

手順の概要

1. **configure terminal**
2. [**no**]**router ospf** *instance-tag*
3. （任意） **router-id** *ip-address*
4. （任意） **show ip ospf** *instance-tag*
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例：	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	`switch# configure terminal` `switch(config)#`	

Step 2	[**no**]**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 3	（任意） **router-id** *ip-address* 例： `switch(config-router)# router-id 192.0.2.1`	OSPFv2 ルータ ID を設定します。この IP アドレスにより、この OSPFv2 インスタンスが識別されます。このアドレスは、システムの設定済みインターフェイス上に存在する必要があります。
Step 4	（任意） **show ip ospf** *instance-tag* 例： `switch(config-router)# show ip ospf 201`	OSPF 情報を表示します。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

OSPFv2インスタンスと、関連付けられている設定をすべて削除するには、グローバルコンフィギュレーション モードで no feature ospf コマンドを使用します。

コマンド	目的
no router ospf *instance-tag* 例： `switch(config)# no router ospf 201`	OSPFインスタンスと、関連付けられた設定を削除します。

（注） このコマンドは、インターフェイスモードではOSPF設定を削除しません。インターフェイスモードで設定されたOSPFv2コマンドはいずれも、手動で削除する必要があります。

OSPFv2 インスタンスのオプション パラメータの設定

OSPF のオプション パラメータを設定できます。高度なOSPFv2の設定（136 ページ）セクションを参照してください。

ルータ コンフィギュレーション モードで、次の OSPFv2 用オプション パラメータを設定できます。

始める前に

OSPF 機能を有効にしてあることを確認します（「OSPFv2 の有効化」の項を参照）。

OSPFv2 がルータ ID（設定済みのループバック アドレスなど）を入手可能であるか、またはルータ ID オプションを設定する必要があります。

手順の概要

1. **distance** *number*
2. **log-adjacency-changes [detail]**
3. **maximum-paths** *path-number*
4. **distance** *number*
5. **log-adjacency-changes** [**detail**]
6. **maximum-paths** *path-number*
7. **passive-interface default**
8. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**distance** *number* 例： `switch(config-router)# distance 25`	この OSPFv2 インスタンスのアドミニストレーティブ ディスタンスを設定します。範囲は 1 ～ 255 です。デフォルトは 110 です。
Step 2	**log-adjacency-changes [detail]** 例： `switch(config-router)# log-adjacency-changes`	ネイバーの状態が変化するたびに、システム メッセージを生成します。
Step 3	**maximum-paths** *path-number* 例： `switch(config-router)# maximum-paths 4`	ルート テーブル内の宛先への同じ OSPFv2 パスの最大数を設定します。このコマンドはロード バランシングに使用されます。指定できる範囲は 1 ～ 16 です。デフォルト値は 8 です。
Step 4	**distance** *number* 例： `switch(config-router)# distance 25`	この OSPFv2 インスタンスのアドミニストレーティブ ディスタンスを設定します。範囲は 1 ～ 255 です。デフォルトは 110 です。
Step 5	**log-adjacency-changes** [**detail**] 例：	ネイバーの状態が変化するたびに、システム メッセージを生成します。

	コマンドまたはアクション	目的
	`switch(config-router)# log-adjacency-changes`	
Step 6	**maximum-paths** *path-number* 例： `switch(config-router)# maximum-paths 4`	ルート テーブル内の宛先への同じ OSPFv2 パスの最大数を設定します。このコマンドはロード バランシングに使用されます。指定できる範囲は 1 ～ 16 です。デフォルト値は 8 です。
Step 7	**passive-interface default** 例： `switch(config-router)# passive-interface default`	すべてのインターフェイス上でルーティングが更新されないようにします。このコマンドは、VRF またはインターフェイス コマンド モードの設定によって上書きされます。
Step 8	（任意） **copy running-config startup-config** 例： `switch(config-router)# copy running-config startup-config`	この設定変更を保存します。

例

次の例は、OSPFv2 インスタンスを作成する方法を示しています。

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

OSPFv2でのネットワークの設定

ルータがこのネットワークへの接続に使用するインターフェイスを介して、OSPFv2 へのネットワークを関連付けることで、このネットワークを設定できます（「ネイバー」セクションを参照）。すべてのネットワークをデフォルト バックボーン エリア（エリア 0）に追加したり、任意の 10 進数または IP アドレスを使用して新規エリアを作成したりできます。

（注） すべてのエリアは、バックボーン エリアに直接、または仮想リンク経由で接続する必要があります。

（注） インターフェイスに有効な IP アドレスを設定するまでは、OSPF はインターフェイス上でイネーブルにされません。

始める前に

OSPF 機能をイネーブルにしてあることを確認します（「OSPFv2 のイネーブル化」の項を参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ip address** *ip-prefix/length*
4. **ip router ospf** *instance-tag* **area** *area-id* [**secondaries none**]
5. （任意） **show ip ospf** *instance-tag* **interface** *interface-type slot/port*
6. **copy running-config startup-config**
7. （任意） **ip ospf cost** *number*
8. （任意） **ip ospf dead-interval** *seconds*
9. （任意） **ip ospf hello-interval** *seconds*
10. （任意） **ip ospf mtu-ignore**
11. （任意） **[default | no] ip ospf passive-interface**
12. （任意） **ip ospf priority** *number*
13. （任意） **ip ospf shutdown**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**ip address** *ip-prefix/length* 例： `switch(config-if)# ip address 192.0.2.1/16`	このインターフェイスにIPアドレスおよびサブネット マスクを割り当てます。
Step 4	**ip router ospf** *instance-tag* **area** *area-id* [**secondaries none**] 例： `switch(config-if)# ip router ospf 201 area 0.0.0.15`	OSPFv2 インスタンスおよびエリアにインターフェイスを追加します。

	コマンドまたはアクション	目的
Step 5	（任意） **show ip ospf** *instance-tag* **interface** *interface-type slot/port* 例： `switch(config-if)# show ip ospf 201 interface ethernet 1/2`	OSPF 情報を表示します。
Step 6	**copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。
Step 7	（任意） **ip ospf cost** *number* 例： `switch(config-if)# ip ospf cost 25`	このインターフェイスの OSPFv2 コストメトリックを設定します。デフォルトでは、参照帯域幅とインターフェイス帯域幅に基づいて、コストメトリックが計算されます。有効な範囲は 1 ～ 65535 です。
Step 8	（任意） **ip ospf dead-interval** *seconds* 例： `switch(config-if)# ip ospf dead-interval 50`	OSPFv2 デッド間隔を秒単位で設定します。有効な範囲は 1 ～ 65535 です。デフォルトでは、hello 間隔の秒数の 4 倍です。
Step 9	（任意） **ip ospf hello-interval** *seconds* 例： `switch(config-if)# ip ospf hello-interval 25`	OSPFv2 hello 間隔を秒単位で設定します。有効な範囲は 1 ～ 65535 です。デフォルトは 10 秒です。
Step 10	（任意） **ip ospf mtu-ignore** 例： `switch(config-if)# ip ospf mtu-ignore`	OSPFv2 で、ネイバーとのあらゆる IP MTU 不一致が無視されるように設定します。デフォルトでは、ネイバー MTU がローカル インターフェイス MTU が不一致の場合には、隣接関係が確立されません。
Step 11	（任意） **[default | no] ip ospf passive-interface** 例： `switch(config-if)# ip ospf passive-interface`	インターフェイス上でルーティングが更新されないようにします。このコマンドによって、ルータまたは VRF コマンド モードの設定が上書きされます。**default** オプションは、このインターフェイス モード コマンドを削除して、ルータまたは VRF の設定に戻します（設定がある場合）。
Step 12	（任意） **ip ospf priority** *number* 例： `switch(config-if)# ip ospf priority 25`	エリアの DR の決定に使用される OSPFv2 プライオリティを設定します。有効な範囲は 0 ～ 255 です。デフォルトは1です。「指定ルータ（172 ページ）」の項を参照してください。
Step 13	（任意） **ip ospf shutdown** 例：	このインターフェイス上の OSPFv2 インスタンスをシャットダウンします。

	コマンドまたはアクション	目的
	`switch(config-if)# ip ospf shutdown`	

例

次に、OSPFv2 インスタンス 201 にネットワーク エリア 0.0.0.10 を追加する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
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
```

show ip ospf interface コマンドを使用し、 すれば、インターフェイスの設定を確認できます。**show ip ospf neighbor** コマンドを使用し、 すれば、このインターフェイスのNAVER を確認できます。

エリアの認証の設定

エリア内のすべてのネットワーク、またはエリア内の個々のインターフェイスの認証を設定できます。インターフェイス認証設定を使用すると、エリア認証は無効になります。

始める前に

OSPF 機能が有効になっていることを確認するには、「OSPFv2 の有効化」セクションを参照してください。

インターフェイス上のすべてのネイバーが、共有認証キーを含め、同じ認証設定を共有することを確認します。

この認証設定のためのキー チェーンを作成します。『Cisco Nexus 9000 シリーズ NX-OS セキュリティ設定ガイド』を参照してください。

（注） OSPFv2 の場合、**key** *key-id* にキー ID があります コマンドは、0 ー 255 の値のみをサポートします。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **area** *area-id* **authentication** [**message-digest**]
4. **interface** *interface-type slot/port*
5. （任意） **ip ospf authentication-key** [**0** | **3**] *key*
6. （任意） **ip ospf message-digest-key** *key-id* **md5** [**0** | **3**] *key*
7. （任意） **show ip ospf** *instance-tag* **interface** *interface-type slot/port*

8. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 3	**area** *area-id* **authentication** [**message-digest**] 例： `switch(config-router)# area 0.0.0.10 authentication`	エリアの認証モードを設定します。

Step 4	**interface** *interface-type slot/port*
	例：
	`switch(config-router)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 5	（任意） **ip ospf authentication-key** [**0** | **3**] *key* 例： `switch(config-if)# ip ospf authentication-key 0 mypass`	このインターフェイスに簡易パスワード認証を設定します。認証が、キーチェーンにもメッセージ ダイジェストにも設定されていない場合は、このコマンドを使用します。0 の場合は、パスワードをクリア テキストで設定します。3 の場合は、パスワードを 3DES 暗号化として設定します。
Step 6	（任意） **ip ospf message-digest-key** *key-id* **md5** [**0** | **3**] *key* 例： `switch(config-if)# ip ospf message-digest-key 21 md5 0 mypass`	このインターフェイスにメッセージ ダイジェスト認証を設定します。認証がメッセージ ダイジェストに設定されている場合は、このコマンドを使用します。key-id の範囲は 1 ～ 255 です。MD5 オプションが 0 の場合はパスワードがクリア テキストで設定され、3 の場合はパス キーが 3DES 暗号化として設定されます。
Step 7	（任意） **show ip ospf** *instance-tag* **interface** *interface-type slot/port* 例：	OSPF 情報を表示します。

	コマンドまたはアクション	目的
	`switch(config-if)# show ip ospf 201 interface ethernet 1/2`	
Step 8	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

インターフェイスの認証の設定

エリア内のすべてのネットワーク、またはエリア内の個々のインターフェイスの認証を設定できます。インターフェイス認証設定を使用すると、エリア認証は無効になります。

始める前に

OSPF 機能をイネーブルにしてあることを確認します（「OSPFv2 のイネーブル化」の項を参照）。

インターフェイス上のすべてのネイバーが、共有認証キーを含め、同じ認証設定を共有することを確認します。

この認証設定のためのキー チェーンを作成します。『Cisco Nexus 9000 シリーズ NX-OS セキュリティ設定ガイド』を参照してください。

（注） OSPFv2 の場合、**key** *key-id* にキー ID がありますコマンドは、0 ー 255 の値のみをサポートします。

キーチェーンでは、キー 0 ～ 255 のみが OSPFv2 でサポートされます。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ip ospf authentication** [**message-digest**]
4. （任意） **ip ospf authentication key-chain** *key-id*
5. （任意） **ip ospf authentication-key** [**0** | **3** | **7**] *key*
6. （任意） **ip ospf message-digest-key** *key-id* **md5** [**0** | **3** | **7**] *key*
7. （任意） **show ip ospf** *instance-tag* **interface** *interface-type slot/port*
8. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**ip ospf authentication** [**message-digest**] 例： `switch(config-if)# ip ospf authentication`	OSPFv2 のインターフェイス認証モードをクリアテキストタイプとメッセージダイジェストタイプのどちらかでイネーブルにします。これにより、エリアに基づくこのインターフェイスの認証が無効となります。すべてのネイバーが、この認証タイプを共有する必要があります。
Step 4	（任意） **ip ospf authentication key-chain** *key-id* 例： `switch(config-if)# ip ospf authentication key-chain Test1`	OSPFv2 のキーチェーンを使用するようにインターフェイス認証を設定します。キー チェーンの詳細については、『*Cisco NX-OSCisco NX-OS* セキュリティ設定ガイド』を参照してください。
Step 5	（任意） **ip ospf authentication-key** [**0** | **3** | **7**] *key* 例： `switch(config-if)# ip ospf authentication-key 0 mypass`	このインターフェイスに簡易パスワード認証を設定します。認証が、キーチェーンにもメッセージ ダイジェストにも設定されていない場合は、このコマンドを使用します。 オプションは次のとおりです。 • 0：パスワードをクリアテキストで設定します。 • 3：パス キーを 3DES 暗号化として設定します。 • 7：パス キーを Cisco タイプ 7 暗号化として設定します。
Step 6	（任意） **ip ospf message-digest-key** *key-id* **md5** [**0** | **3** | **7**] *key* 例： `switch(config-if)# ip ospf message-digest-key 21 md5 0 mypass`	このインターフェイスにメッセージ ダイジェスト認証を設定します。認証がメッセージ ダイジェストに設定されている場合は、このコマンドを使用します。key-id の範囲は 1 ～ 255 です。MD5 オプションは次のとおりです。 • 0：パスワードをクリアテキストで設定します。

	コマンドまたはアクション	目的
		• 3：パス キーを 3DES 暗号化として設定します。 • 7：パス キーを Cisco タイプ 7 暗号化として設定します。
Step 7	（任意） **show ip ospf** *instance-tag* **interface** *interface-type slot/port* 例： `switch(config-if)# show ip ospf 201 interface ethernet 1/2`	OSPF 情報を表示します。
Step 8	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、インターフェイスに暗号化されていない簡単なパスワードを設定し、イーサネットインターフェイス 1/2 のパスワードを設定する例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# exit
switch(config)# interface ethernet 1/2
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
```

次に、OSPFv2 HMAC-SHA-1 および MD5 暗号化認証を設定する例を示します。

```
switch# configure terminal
switch(config)# key chain chain1
switch(config-keychain)# key 1
switch(config-keychain-key)# key-string 7 070724404206
switch(config-keychain-key)# accept-lifetime 01:01:01 Jan 01 2015 infinite
switch(config-keychain-key)# send-lifetime 01:01:01 Jan 01 2015 infinite
switch(config-keychain-key)# cryptographic-algorithm HMAC-SHA-1
switch(config-keychain-key)# exit
switch(config-keychain)# key 2
switch(config-keychain-key)# key-string 7 070e234f1f5b4a
switch(config-keychain-key)# accept-lifetime 10:51:01 Jul 24 2015 infinite
switch(config-keychain-key)# send-lifetime 10:51:01 Jul 24 2015 infinite
switch(config-keychain-key)# cryptographic-algorithm MD5
switch(config-keychain-key)# exit
switch(config-keychain)# exit

switch(config)# interface ethernet 1/1
switch(config-if)# ip router ospf 1 area 0.0.0.0
switch(config-if)# ip ospf authentication message-digest
switch(config-if)# ip ospf authentication key-chain chain1
```

```
switch(config-if)# show key chain chain1
Key-Chain chain1
Key 1 -- text 7 "070724404206"
cryptographic-algorithm HMAC-SHA-1
accept lifetime UTC (01:01:01 Jan 01 2015)-(always valid) [active]
send lifetime UTC (01:01:01 Jan 01 2015)-(always valid) [active]
Key 2 -- text 7 "070e234f1f5b4a"
cryptographic-algorithm MD
accept lifetime UTC (10:51:00 Jul 24 2015)-(always valid) [active]
send lifetime UTC (10:51:00 Jul 24 2015)-(always valid) [active]

switch(config-if)# show ip ospf interface ethernet 1/1
Ethernet1/1 is up, line protocol is up
IP address 11.11.11.1/24
Process ID 1 VRF default, area 0.0.0.3
Enabled by interface configuration
State BDR, Network type BROADCAST, cost 40
Index 6, Transmit delay 1 sec, Router Priority 1
Designated Router ID: 33.33.33.33, address: 11.11.11.3
Backup Designated Router ID: 1.1.1.1, address: 11.11.11.1
2 Neighbors, flooding to 2, adjacent with 2
Timer intervals: Hello 10, Dead 40, Wait 40, Retransmit 5
Hello timer due in 00:00:08
Message-digest authentication, using keychain key1 (ready)
Sending SA: Key id 2, Algorithm MD5
Number of opaque link LSAs: 0, checksum sum 0
```

高度なOSPFv2の設定

OSPFv2 は、OSPFv2 ネットワークを設計した後に設定します。

境界ルータのフィルタ リストの設定

OSPFv2 ドメインを関連ネットワークを含む一連のエリアに分割できます。すべてのエリアは、エリア境界ルータ（ABR）経由でバックボーン エリアに接続している必要があります。OSPFv2 ドメインは、自律システム境界ルータ（ASBR）を介して、外部ドメインにも接続可能です。「エリア （173 ページ）」の項を参照してください。

ABR には、省略可能な次の設定パラメータがあります。

- Area range：エリア間のルート集約を設定します。「ルート集約の設定」の項を参照してください。
- Filter list：外部エリアから受信したネットワーク集約（タイプ 3）LSA をフィルタリングします。

ASBR もフィルタ リストをサポートしています。

始める前に

OSPF 機能がイネーブルになっていることを確認します。「OSPFv2 の有効化」の項を参照してください。

フィルタ リストが、着信または発信ネットワーク集約（タイプ 3）LSA の IP プレフィックスのフィルタリングに使用するルート マップを作成します。「Route Policy Manager の設定」を参照してください。「エリア （173 ページ）」を参照してください。

手順の概要

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

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 3	**area** *area-id* **filter-list route-map** *map-name* {**in**	**out**} 例： `switch(config-router)# area 0.0.0.10 filter-list route-map FilterLSAs in`	ABR 上で着信または発信ネットワーク集約（タイプ 3）LSA をフィルタリングします。
Step 4	（任意） **show ip ospf policy statistics area** *id* **filter-list** {**in**	**out**} 例： `switch(config-router)# show ip ospf policy statistics area 0.0.0.10 filter-list in`	OSPF ポリシー情報を表示します。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、エリア 0.0.0.10 でフィルタ リストを設定する例を示します。

```
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
```

スタブ エリアの設定

OSPFv2 ドメインの外部トラフィックが不要な個所にスタブ エリアを設定できます。スタブ エリアはAS外部（タイプ5）LSAをブロックし、選択したネットワークへの往復の不要なルーティングを制限します。「スタブ エリア」の項を参照してください。また、すべての集約ルートがスタブ エリアを経由しないようブロックすることもできます。

始める前に

OSPF 機能がイネーブルになっていることを確認します。（「OSPFv2 の有効化」の項を参照）。

設定されるスタブ エリア内に、仮想リンクと ASBR のいずれも含まれないことを確認します。

手順の概要

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

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

	コマンドまたはアクション	目的
Step 3	**area** *area-id* **stub** 例： `switch(config-router)# area 0.0.0.10 stub`	このエリアをスタブ エリアとして作成します。
Step 4	（任意） **area** *area-id* **default-cost** *cost* 例： `switch(config-router)# area 0.0.0.10 default-cost 25`	このスタブ エリアに送信されるデフォルト サマリ ルートのコスト メトリックを設定します。指定できる範囲は 0 ～ 16777215 です。デフォルトは 1 です。
Step 5	（任意） **show ip ospf** *instance-tag* 例： `switch(config-router)# show ip ospf 201`	OSPF 情報を表示します。
Step 6	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、スタブ エリアを作成する例を示します。

```
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
```

Totally Stubby エリアの設定

Totally Stubby エリアを作成して、すべての集約ルート更新がスタブ エリアに入るのを防ぐことができます。

Totally Stubby エリアを作成するには、ルータ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **area** *area-id* **stub no-summary**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**area** *area-id* **stub no-summary** 例： `switch(config-router)# area 20 stub no-summary`	このエリアを Totally Stubby エリアとして作成します。

NSSA の設定

OSPFv2 ドメインの一部で一定限度の外部トラフィックが必要な場合は、その部分に NSSA を設定できます。また、この外部トラフィックを AS 外部（タイプ 5）LSA に変換して、このルーティング情報で OSPFv2 ドメインをフラッディングすることもできます。NSSA は、省略可能な次のパラメータで設定できます。

- No redistribution：再配布されたルートは、NSSA をバイパスして OSPFv2 自律システム内の他のエリアに再配布されます。このオプションは、NSSA ASBR が ABR も兼ねているときに使用します。
- Default information originate：外部自律システムへのデフォルト ルートの NSSA 外部（タイプ 7）LSA を生成します。このオプションは、ASBR のルーティング テーブルにデフォルト ルートが含まれる場合に NSSA ASBR 上で使用します。このオプションは、ASBR のルーティング テーブルにデフォルト ルートが含まれるかどうかに関係なく、NSSA ASBR 上で使用できます。
- Route map：目的のルートだけが NSSA および他のエリア全体でフラッディングされるように、外部ルートをフィルタリングします。
- No summary：すべての集約ルートが NSSA でフラッディングされないようにします。このオプションは NSSA ABR 上で使用します。
- Translate：NSSA 外のエリア向けに、NSSA 外部 LSA を AS 外部 LSA に変換します。再配布されたルートを OSPFv2 自律システム全体でフラッディングするには、このコマンドを NSSA ABR 上で使用します。また、これらの AS 外部 LSA の転送アドレスを無効にすることもできます。このオプションを選択した場合は、転送アドレスが 0.0.0.0 に設定されます。

（注） 変換オプションでは、NSSA を作成し、他のオプションを設定する **area** *area-id* **nssa** コマンドの後に、別の **area** *area-id* **nssa** コマンドが必要です。

（注） 変換を有効にするには、コマンド area 0.0.0.2 NSSA translation type7 を使用します。エリア 2 を NSSA として指定するように、コマンド area 0.0.0.2 NSSA を設定していることを確認します。

始める前に

OSPF 機能を有効にしてあることを確認します（「OSPFv2 の有効化」の項を参照）。

設定する NSSA 上に仮想リンクがないことと、この NSSA がバックボーン エリアでないことを確認します。

手順の概要

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

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 3	**area** *area-id* **nssa [no-redistribution] [default-information-originate]originate** [**route-map** *map-name*]] [**no-summary**] 例： `switch(config-router)# area 0.0.0.10 nssa no-redistribution`	このエリアを NSSA として作成します。

	コマンドまたはアクション	目的
Step 4	（任意） **area** *area-id* **nssa translate type7** {**always** | **never**} [**suppress-fa**] 例： `switch(config-router)# area 0.0.0.10 nssa translate type7 always`	AS 外部（タイプ 7）LSA を NSSA 外部（タイプ 5）LSA に変換するように NSSA を設定します。
Step 5	（任意） **area** *area-id* **default-cost** *cost* 例： `switch(config-router)# area 0.0.0.10 default-cost 25`	この NSSA に送信されるデフォルト集約ルートのコスト メトリックを設定します。
Step 6	（任意） **show ip ospf** *instance-tag* 例： `switch(config-router)# show ip ospf 201`	OSPF 情報を表示します。
Step 7	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、すべての集約ルート更新をブロックする NSSA を作成する例を示します。

```
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
```

次に、デフォルト ルートを生成する NSSA を作成する例を示します。

```
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
```

次に、外部ルートをフィルタリングし、すべての集約ルート更新をブロックする NSSA を作成する例を示します。

```
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
```

次に、常に NSSA 外部（タイプ 5）LSA を AS 外部（タイプ 7）LSA に変換する NSSA を作成し NSSA を設定する例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 nssa
```

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

マルチエリアの隣接関係の設定

既存の OSPFv2 インターフェイスには複数のエリアを追加できます。追加の論理インターフェイスはマルチエリア隣接関係をサポートしています。

始める前に

OSPFv2 機能が有効にされている必要があります（「OSPFv2 の有効化」のセクションを参照）。

インターフェイスにプライマリ エリアが設定されていることを確認します（「OSPFv2 のネットワークの設定」の項を参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ip router ospf** [*instance-tag*] **multi-area** *area-id*
4. （任意） **show ip ospf** *instance-tag* **interface** *interface-type slot/port*
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**ip router ospf** [*instance-tag*] **multi-area** *area-id* 例： `switch(config-if)# ip router ospf 201 multi-area 3`	別のエリアにインターフェイスを追加します。 （注） Cisco NX-OS リリース 7.0(3)I5(1) 以降では、*instance-tag* 引数はオプションです。インスタンスを指定しない場合、マルチエリア設定は、そのインターフェイスのプライマリ エリアに設定されている同じインスタンスに適用されます。

	コマンドまたはアクション	目的
Step 4	（任意） **show ip ospf** *instance-tag* **interface** *interface-type slot/port* 例： `switch(config-if)# show ip ospf 201 interface ethernet 1/2`	OSPFv2 情報を表示します。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、OSPFv2 インターフェイスに別のエリアを追加する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
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)# ip router ospf 201 multi-area 20
switch(config-if)# copy running-config startup-config
```

仮想リンクの設定

仮想リンクは、隔離されたエリアを中継エリアを介してバックボーン エリアに接続します。「仮想リンク」の項を参照してください。仮想リンクには、省略可能な次のパラメータを設定できます。

- Authentication：簡単なパスワード認証または MD5 メッセージ ダイジェスト認証、および関連付けられたキーを設定します。
- Dead interval：ローカル ルータがデッドであることを宣言し、隣接関係を解消する前に、ネイバーが hello パケットを待つ時間を設定します。
- Hello interval：連続する hello パケット間の時間間隔を設定します。
- Retransmit interval：連続する LSA 間の推定時間間隔を設定します。
- Transmit delay：LSA をネイバーに送信する推定時間を設定します。

（注） リンクがアクティブになる前に、関与する両方のルータで仮想リンクを設定する必要があります。

スタブ エリアには仮想リンクを追加できません。

始める前に

OSPF 機能をイネーブルにしてあることを確認します（「OSPFv2 のイネーブル化」の項を参照）。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **area** *area-id* **virtual link** *router-id*
4. （任意） **show ip ospf virtual-link** [**brief**]
5. （任意） **copy running-config startup-config**
6. （任意） **authentication** [**key-chain** *key-id* **message-digest** | **null**]
7. （任意） **authentication-key** [**0** | **3**] *key*
8. （任意） **dead-interval** *seconds*
9. （任意） **hello-interval** *seconds*
10. （任意） **message-digest-key** *key-id* **md5** [**0** | **3**] *key*
11. （任意） **retransmit-interval** *seconds*
12. （任意） **transmit-delay** *seconds*

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 3	**area** *area-id* **virtual link** *router-id*
	例：
	`switch(config-router)# area 0.0.0.10 virtual-link 10.1.2.3`
	`switch(config-router-vlink)#`
		リモートルータへの仮想リンクの端を作成します。仮想リンクをリモートルータ上に作成して、リンクを完成させる必要があります。

Step 4	（任意） **show ip ospf virtual-link** [**brief**] 例： `switch(config-router-vlink)# show ip ospf virtual-link`	OSPF 仮想リンク情報を表示します。

	コマンドまたはアクション	目的
Step 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップコンフィギュレーションにコピーします
Step 6	（任意） **authentication** [**key-chain** *key-id* **message-digest** | **null**] 例： `switch(config-router-vlink)# authentication message-digest`	エリアに基づくこの仮想リンクの認証がオーバーライドされます。
Step 7	（任意） **authentication-key** [**0** | **3**] *key* 例： `switch(config-router-vlink)# authentication-key 0 mypass`	この仮想リンクに簡易パスワードを設定します。認証が、キーチェーンにもメッセージダイジェストにも設定されていない場合は、このコマンドを使用します。0 の場合は、パスワードをクリア テキストで設定します。3 の場合は、パスワードを 3DES 暗号化として設定します。
Step 8	（任意） **dead-interval** *seconds* 例： `switch(config-router-vlink)# dead-interval 50`	OSPFv2 デッド間隔を秒単位で設定します。有効な範囲は 1 ～ 65535 です。デフォルトでは、hello 間隔の秒数の 4 倍です。
Step 9	（任意） **hello-interval** *seconds* 例： `switch(config-router-vlink)# hello-interval 25`	OSPFv2 hello 間隔を秒単位で設定します。有効な範囲は 1 ～ 65535 です。デフォルトは 10 秒です。
Step 10	（任意） **message-digest-key** *key-id* **md5** [**0** | **3**] *key* 例： `switch(config-router-vlink)# message-digest-key 21 md5 0 mypass`	この仮想リンクにメッセージダイジェスト認証を設定します。認証がメッセージダイジェストに設定されている場合は、このコマンドを使用します。0 の場合は、パスワードをクリア テキストで設定します。3 の場合は、パス キーを 3DES 暗号化として設定します。
Step 11	（任意） **retransmit-interval** *seconds* 例： `switch(config-router-vlink)# retransmit-interval 50`	OSPFv2 再送信間隔を秒単位で設定します。有効な範囲は 1 ～ 65535 です。デフォルトは 5 分です。
Step 12	（任意） **transmit-delay** *seconds* 例： `switch(config-router-vlink)# transmit-delay 2`	OSPFv2 送信遅延を秒単位で設定します。指定できる範囲は 1 ～ 450 です。デフォルトは 1 です。

例

（注） OSPFv2 の場合、key key-id コマンドのキー ID の値は 0 ～ 255 です。

キーチェーンでは、キー 0 ～ 255 のみが OSPFv2 でサポートされます。

次に、2 つの ABR 間に簡単な仮想リンクを作成する例を示します。

ABR 1（ルータ ID 27.0.0.55）の設定は、次のとおりです。

```
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)# copy running-config startup-config
```

ABR 2（ルータ ID 10.1.2.3）の設定は、次のとおりです。

```
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)# copy running-config startup-config
```

再配布の設定

他のルーティング プロトコルから学習したルートを、ASBR 経由で OSPFv2 自律システムに再配布できます。

デフォルトルートを再配布するには、次のパラメータを指定する必要があります。

- 5**default-information originate**：デフォルト ルートが RIB に存在する場合は、この OSPF ドメインにデフォルト ルートを作成します。

（注） Cisco NX-OS リリース 7.0(3)I7(6) 以降では、デフォルト ルートを OSPF に再配布する場合、Cisco NX-OS はデフォルト ルートを正常にアドバタイズするために **default-information originate** コマンドを必要とします。

デフォルト以外のルートの場合、OSPF でのルート再配布には、省略可能な次のパラメータを設定できます。

- **default-metric**：すべての再配布ルートに同じコスト メトリックを設定します。

始める前に

OSPF 機能をイネーブルにします。「OSPFv2 の有効化」を参照してください。

再配布で使用する、必要なルート マップを作成します。

手順の概要

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

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 3	**redistribute** {**bgp** *id* | **direct** | **eigrp** *id* | **isis** *id* | **ospf** *id* | **rip** *id* | **static**} **route-map** *map-name* 例： `switch(config-router)# redistribute bgp route-map FilterExternalBGP`	設定したルート マップ経由で、選択したプロトコルを OSPF に再配布します。 （注） Cisco NX-OS リリース 7.0(3)I7(6) 以降では、デフォルト ルートを OSPF に再配布する場合、Cisco NX-OS はデフォルト ルートを正常にアドバタイズするために **default-information originate** コマンドを必要とします。
Step 4	**default-information originate** [**always**] [**route-map** *map-name*] 例： `switch(config-router)# default-information-originate route-map DefaultRouteFilter`	デフォルト ルートが RIB に存在する場合は、この OSPF ドメインにデフォルト ルートを作成します。次の省略可能なキーワードを使用します。 • **always**：常に 0.0.0. のデフォルト ルートを生成します。ルートが RIB に存在しない場合でも。 • **route-map**：ルート マップが true を返す場合にデフォルト ルートを生成します。 （注） このコマンドは、ルート マップの **match** 文を無視します。

	コマンドまたはアクション	目的
Step 5	**default-metric** [*cost*] 例： `switch(config-router)# default-metric 25`	再配布されたルートのコスト メトリックを設定します。このコマンドは、直接接続されたルートには適用されません。ルート マップを使用して、直接接続されたルートのデフォルトのメトリックを設定します。
Step 6	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、ボーダー ゲートウェイ プロトコル（BGP）を OSPF に再配布する例を示します。

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

再配布されるルート数の制限

ルートの再配布によって、OSPFv2 ルート テーブルに多くのルートが追加される可能性があります。外部プロトコルから受け取るルートの数の上限を設定できます。OSPFv2 には、再配布ルートの制限を設定するために次のオプションが用意されています。

- 上限固定：設定された最大値にOSPFv2が達すると、メッセージをログに記録します。OSPFv2 は以降の再配布ルートを受け取りません。任意で、最大値のしきい値パーセンテージを設定して、OSPFv2 がこのしきい値を超えたときに警告を記録するようにすることもできます。
- 警告のみ：OSPFv2 が最大値に達したときのみ、警告のログを記録します。OSPFv2 は、再配布されたルートを受け入れ続けます。
- 取り消し：OSPFv2が最大値に達したときにタイムアウト期間を開始します。このタイムアウト期間後、現在の再配布されたルート数が最大制限より少なければ、OSPFv2はすべての再配布されたルートを要求します。再配布されたルートの現在数が最大数に達した場合、OSPFv2 はすべての再配布されたルートを取り消します。OSPFv2が追加の再配布されたルートを受け付ける前に、この状況を解消する必要があります。
- 任意で、タイムアウト期間を設定できます。

始める前に

OSPF 機能を有効にしてあることを確認します（「OSPFv2 の有効化」の項を参照）。

手順の概要

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

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 3	**redistribute** {**bgp** *id* | **direct** | **eigrp** *id* | **isis** *id* | **ospf** *id* | **rip** *id* | **static**} **route-map** *map-name* 例： `switch(config-router)# redistribute bgp route-map FilterExternalBGP`	設定したルート マップ経由で、選択したプロトコルを OSPF に再配布します。
Step 4	**redistribute maximum-prefix** *max* [*threshold*] [**warning-only | withdraw** [*num-retries timeout*]] 例： `switch(config-router)# redistribute maximum-prefix 1000 75 warning-only`	OSPFv2 が配布するプレフィックスの最大数を指定します。指定できる範囲は 0 ～ 65536 です。任意で次のオプションを指定します。 • *threshold*：警告メッセージをトリガーする最大プレフィクス数のパーセンテージ。 • **warning-only**：プレフィックスの最大数を超えた場合に警告メッセージを記録します。 • **withdraw**：再配布されたすべてのルートを取り消します。任意で再配布されたルートを取得しようと試みます。*num-retries* の範囲は 1 ～ 12 です。*timeout* の範囲は 60 ～ 600 秒です。デフォルトは 300 秒です。**clear ip ospf redistribution** コマンドは、すべてのルートが取り消された場合に使用します。

	コマンドまたはアクション	目的
Step 5	（任意） **show running-config ospf** 例： `switch(config-router)# show running-config ospf`	OSPFv2 設定を表示します。
Step 6	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、OSPF に再配布されるルートの数を制限する例を示します。

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

ルート集約の設定

集約したアドレス範囲を設定することにより、エリア間ルートのルート集約を設定できます。また、ASBR 上のこれらのルートのサマリ アドレスを設定して、外部の再配布されたルートのルート集約を設定することもできます。詳細については、「ルート集約」を参照してください。

始める前に

OSPF 機能を有効にしてあることを確認します（「OSPFv2 の有効化」の項を参照）。

手順の概要

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*]
5. （任意） **show ip ospf summary-address**
6. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 3	**area** *area-id* **range** *ip-prefix/length* [**no-advertise**] [**cost** *cost*] 例： `switch(config-router)# area 0.0.0.10 range 10.3.0.0/16`	一定の範囲のアドレスのサマリ アドレスを ABR 上に作成します。このサマリ アドレスをネットワーク集約（タイプ 3）LSA にアドバタイズしないようにすることもできます。*cost* の範囲は 0 ～ 16777215 です。
Step 4	**summary-address** *ip-prefix/length* [**no-advertise** | **tag** *tag*] 例： `switch(config-router)# summary-address 10.5.0.0/16 tag 2`	一定の範囲のアドレスのサマリ アドレスを ABR 上に作成します。ルート マップによる再配布で使用できるよう、このサマリ アドレスにタグを割り当てることもできます。
Step 5	（任意） **show ip ospf summary-address** 例： `switch(config-router)# show ip ospf summary-address`	OSPF サマリ アドレスに関する情報を表示します。
Step 6	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、ABR 上のエリア間のサマリ アドレスを作成する例を示します。

```
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
```

次に、ASBR 上のサマリ アドレスを作成する例を示します。

```
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
```

スタブ ルート アドバタイズメントの設定

短期間だけ、このルータ経由の OSPFv2 トラフィックを制限する場合は、スタブ ルート アドバタイズメントを使用します。詳細については、「OSPFv2 スタブ ルータ アドバタイズメント」の項を参照してください。

スタブ ルート アドバタイズメントは、省略可能な次のパラメータで設定できます。

- On startup：指定した宣言期間だけ、スタブ ルート アドバタイズメントを送信します。
- Wait for BGP：BGP がコンバージェンスするまで、スタブ ルート アドバタイズメントを送信します。

（注） ルータの実行コンフィギュレーションがグレースフル シャットダウンを行うよう設定されている場合は、その実行コンフィギュレーションを保存しないでください。保存すると、ルータが、リロード後に最大メトリックをアドバタイズし続けることになります。

始める前に

OSPF 機能を有効にしてあることを確認します（「OSPFv2 の有効化」の項を参照）。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **max-metric router-lsa** [**external-lsa** [*max-metric-value*]] [**include-stub**] [**on-startup** {*seconds* | **wait-for bgp** *tag*}] [**summary-lsa** [*max-metric-value*}]
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的

Step 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 3	**max-metric router-lsa** [**external-lsa** [*max-metric-value*]] [**include-stub**] [**on-startup** {*seconds* | **wait-for bgp** *tag*}] [**summary-lsa** [*max-metric-value*}]] 例： `switch(config-router)# max-metric router-lsa`	OSPFv2 スタブ ルート アドバタイズメントを設定します。
Step 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、起動時にスタブ ルータ アドバタイズメントを、デフォルトの 600 秒間イネーブルにする例を示します。

```
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
```

ルートのアドミニストレーティブ ディスタンスの設定

OSPFv2 によって RIB に追加されるルートのアドミニストレーティブ ディスタンスを設定できます。

アドミニストレーティブ ディスタンスは、ルーティング情報源の信頼性を示す評価基準です。値が高いほど信頼性の評価は低くなります。一般的にルートは、複数のルーティング プロトコルを通じて検出されます。アドミニストレーティブディスタンスは、複数のルーティングプロトコルから学習したルートを区別するために使用されます。最もアドミニストレーティブ ディスタンスが低いルートが IP ルーティング テーブルに組み込まれます。

OSPF は、IPv4 および IPv6 プレフィックスの距離をフィルタリングおよび変更するためのテーブル マップをサポートします。

始める前に

OSPF 機能がイネーブルにされていることを確認してください（「OSPFv2 のイネーブル化」の項を参照）。

「OSPFv2 の注意事項と制限事項」の項にあるこの機能の注意事項と制限事項を参照してください。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. [**no**] **table-map** *map-name*
4. **exit**
5. **route-map** *map-name* [**permit** | **deny**] [*seq*]
6. **match route-type** *route-type*
7. **match ip route-source prefix-list** *name*
8. **match ip address prefix-list** *name*
9. **set distance** *value*
10. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 3	[**no**] **table-map** *map-name* 例： `switch(config-router)# table-map foo`	OSPFv2 ルートを RIB に送信する前に、OSPFv2 ルートをフィルタリングまたは変更するポリシーを設定します。マップ名には最大 63 文字の英数字を入力できます。

Step 4	**exit**
	例：
	`switch(config-router)# exit`
	`switch(config)#`
		ルータ コンフィギュレーション モードを終了します。

Step 5	**route-map** *map-name* [**permit** | **deny**] [*seq*]
	例：
	`switch(config)# route-map foo permit 10`
	`switch(config-route-map)#`
		ルート マップを作成するか、または既存のルートマップに対応するルートマップ設定モードを開始します。ルートマップのエントリを順序付けるには、*seq* を使用します。

	コマンドまたはアクション	目的
		（注） **permit** オプションで、ディスタンスを設定することができます。**deny** オプションを使用すると、デフォルトのディスタンスが適用されます。
Step 6	**match route-type** *route-type* 例： `switch(config-route-map)# match route-type external`	次のルート タイプのいずれかと照合します。 • external：外部ルート（BGP、EIGRP、OSPF タイプ 1 または 2） • inter-area：OSPF エリア間ルート • internal：内部ルート（OSPF エリア内またはエリア間ルートを含む） • intra-area：OSPF エリア内ルート • nssa-external：NSSA 外部ルート（OSPF タイプ 1 または 2） • type-1：OSPF 外部タイプ 1 ルート • type-2：OSPF 外部タイプ 2 ルート
Step 7	**match ip route-source prefix-list** *name* 例： `switch(config-route-map)# match ip route-source prefix-list p1`	1 つまたは複数の IP プレフィックス リストに対して、ルートの IPv4 ルート送信元アドレスまたはルータ ID と照合します。プレフィックス リストは **ip prefix-list** コマンドを使用して作成します。
Step 8	**match ip address prefix-list** *name* 例： `switch(config-route-map)# match ip address prefix-list p1`	1 つまたは複数の IPv4 プレフィックス リストと照合。プレフィックス リストは **ip prefix-list** コマンドを使用して作成します。
Step 9	**set distance** *value* 例： `switch(config-route-map)# set distance 150`	OSPFv2 のルートのアドミニストレーティブ ディスタンスを設定します。範囲は 1 ～ 255 です。
Step 10	（任意） **copy running-config startup-config** 例： `switch(config-route-map)# copy running-config startup-config`	この設定変更を保存します。

例

次に、OSPFv2 アドミニストレーティブ ディスタンスについて、エリア間ルートを 150、外部ルートを 200、およびプレフィックス リスト p1 内のすべてのプレフィックスを 190 に設定する例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# table-map foo
switch(config-router)# exit
switch(config)# route-map foo permit 10
switch(config-route-map)# match route-type inter-area
switch(config-route-map)# set distance 150
switch(config-route-map)# exit
switch(config)# route-map foo permit 20
switch(config-route-map)# match route-type external
switch(config-route-map)# set distance 200
switch(config-route-map)# exit
switch(config)# route-map foo permit 30
switch(config-route-map)# match ip route-source prefix-list p1
switch(config-route-map)# match ip address prefix-list p1
switch(config-route-map)# set distance 190
```

デフォルト タイマーの変更

OSPFv2 には、プロトコル メッセージの動作および最短パス優先（SPF）の計算を制御する多数のタイマーが含まれています。OSPFv2 には、省略可能な次のタイマー パラメータが含まれます。

- LSA arrival time：ネイバーから着信する LSA 間で許容される最小間隔を設定します。この時間より短時間で到着する LSA はドロップされます。
- Pacing LSAs：LSA が集められてグループ化され、リフレッシュされて、チェックサムが計算される間隔、つまり期限切れとなる間隔を設定します。このタイマーは、LSA 更新が実行される頻度を制御し、LSA 更新メッセージで送信される LSA 更新の数を最適化します（フラッディングと LSA グループ ペーシング （176 ページ） セクションを参照）。
- Throttle LSAs：LSA 生成のレート制限を設定します。このタイマーは、トポロジが変更された後に LSA が生成される頻度を制御します。
- Throttle SPF calculation：SPF 計算の実行頻度を制御します。

インターフェイス レベルでは、次のタイマーも制御できます。

- Retransmit interval：連続する LSA 間の推定時間間隔を設定します。
- Transmit delay：LSA をネイバーに送信する推定時間を設定します。

hello 間隔とデッド タイマーに関する情報の詳細については、「OSPFv2 のネットワークの設定」の項を参照してください。

始める前に

OSPF 機能を有効にしてあることを確認します（「OSPFv2 の有効化」の項を参照）。

手順の概要

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-wait*
7. **interface** *type slot/port*
8. **ip ospf hello-interval** *seconds*
9. **ip ospf dead-interval** *seconds*
10. **ip ospf retransmit-interval** *seconds*
11. **ip ospf transmit-delay** *seconds*
12. （任意） **show ip ospf**
13. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 3	**timers lsa-arrival** *msec* 例： `switch(config-router)# timers lsa-arrival 2000`	LSA 到着時間をミリ秒で設定します。範囲は 10 ～ 600000 です。デフォルトは 1000 ミリ秒です。
Step 4	**timers lsa-group-pacing** *seconds* 例： `switch(config-router)# timers lsa-group-pacing 1800`	LSA がグループ化される間隔を秒で設定します。範囲は 1 ～ 1800 です。デフォルトは 240 秒です。

	コマンドまたはアクション	目的
Step 5	**timers throttle lsa** *start-time hold-interval max-time* 例： `switch(config-router)# timers throttle lsa 3000 6000 6000`	次のタイマーを使用して、LSA 生成のレート制限をミリ秒で設定します。 • *start-time*：指定できる範囲は 0 ～ 5000 ミリ秒です。デフォルト値は 0 ミリ秒です。 • *hold-interval*：指定できる範囲は 50 ～ 30,000 ミリ秒です。デフォルト値は 5000 ミリ秒です。 • *max-time*：指定できる範囲は 50 ～ 30,000 ミリ秒です。デフォルト値は 5000 ミリ秒です。
Step 6	**timers throttle spf** *delay-time hold-time max-wait* 例： `switch(config-router)# timers throttle spf 3000 2000 4000`	SPF 最適パス スケジュールを次のタイマを使用して、SPF 最適パス計算間（秒単位）で設定します。 • *delay-time*：範囲は 1 ～ 600000 ミリ秒です。デフォルトは 200 ミリ秒です。 • *hold-time*：範囲は 1 ～ 600000 ミリ秒です。デフォルト値は、1000 ミリ秒です。 • *max-wait* ：範囲は 1 ～ 600000 ミリ秒です。デフォルト値は 5000 ミリ秒です。

Step 7	**interface** *type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)`
		インターフェイス設定モードを開始します。

Step 8	**ip ospf hello-interval** *seconds* 例： `switch(config-if)# ip ospf hello-interval 30`	このインターフェイスの hello 間隔を設定します。有効な範囲は 1 ～ 65535 です。デフォルトは 10 です。
Step 9	**ip ospf dead-interval** *seconds* 例： `switch(config-if)# ip ospf dead-interval 30`	このインターフェイスのデッド間隔を設定します。有効な範囲は 1 ～ 65535 です。
Step 10	**ip ospf retransmit-interval** *seconds* 例： `switch(config-if)# ip ospf retransmit-interval 30`	このインターフェイスから送信される各 LSA 間の推定時間間隔を設定します。有効な範囲は1～65535 です。デフォルトは 5 分です。

	コマンドまたはアクション	目的

Step 11	**ip ospf transmit-delay** *seconds*
	例：
	`switch(config-if)# ip ospf transmit-delay 450`
	`switch(config-if)#`
		LSA をネイバーに送信する推定時間間隔を秒で設定します。指定できる範囲は 1 ～ 450 です。デフォルトは 1 です。

Step 12	（任意） **show ip ospf** 例： `switch(config-if)# show ip ospf`	OSPF に関する情報を表示します。
Step 13	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップコンフィギュレーションにコピーします

例

次に、lsa-group-pacing オプションで LSA フラッディングを制御する例を示します。

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

グレースフル リスタートの設定

デフォルトでは、グレースフル リスタートは有効です。OSPFv2 インスタンスのグレースフル リスタートには、省略可能な次のパラメータを設定できます。

- Grace period：グレースフル リスタートの開始後に、ネイバーが隣接関係を解消するまでに待つ時間を設定します。
- Helper mode disabled：ローカル OSPFv2 インスタンスのヘルパー モードを無効にします。OSPFv2 は、ネイバーのグレースフル リスタートには関与しません。
- Planned graceful restart only：予定された再起動の場合にだけグレースフル リスタートがサポートされるように OSPFv2 を設定します。

始める前に

OSPF 機能が有効にされていることを確認してください（「OSPFv2 の有効化」のセクションを参照）。

すべてのネイバーで、一致した省略可能なパラメータ一式とともにグレースフル リスタートが設定されていることを確認します。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **graceful-restart**
4. （任意） **graceful-restart grace-period** *seconds*
5. （任意） **graceful-restart helper-disable**
6. （任意） **graceful-restart planned-only**
7. （任意） **show ip ospf** *instance-tag*
8. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 3	**graceful-restart** 例： `switch(config-router)# graceful-restart`	グレースフル リスタートを有効にします。グレースフル リスタートは、デフォルトで有効にされています。
Step 4	（任意） **graceful-restart grace-period** *seconds* 例： `switch(config-router)# graceful-restart grace-period 120`	猶予期間を秒で設定します。指定できる範囲は 5 ～ 1800 です。デフォルトは 60 秒です。
Step 5	（任意） **graceful-restart helper-disable** 例： `switch(config-router)# graceful-restart helper-disable`	ヘルパー モードを無効にします。この機能はデフォルトで有効になっています。
Step 6	（任意） **graceful-restart planned-only** 例： `switch(config-router)# graceful-restart planned-only`	予定された再起動時にのみグレースフル リスタートを設定します。

	コマンドまたはアクション	目的
Step 7	（任意） **show ip ospf** *instance-tag* 例： `switch(config-router)# show ip ospf 201`	OSPF 情報を表示します。
Step 8	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、ディセーブルにされているグレースフル リスタートをイネーブルにし、猶予期間を 120 秒に設定する例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# graceful-restart
switch(config-router)# graceful-restart grace-period 120
switch(config-router)# copy running-config startup-config
```

OSPFv2 インスタンスの再起動

OSPv2インスタンスを再起動できます。この処理では、インスタンスのすべてのネイバーが消去されます。

OSPFv2インスタンスを再起動して、関連付けられたすべてのネイバーを削除するには、次のコマンドを使用します。

手順の概要

1. **restart ospf** *instance-tag*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**restart ospf** *instance-tag* 例： `switch(config)# restart ospf 201`	OSPFv2 インスタンスを再起動して、すべてのネイバーを削除します。

仮想化による OSPFv2 の設定

複数の OSPFv2 インスタンスを作成することができます。また、複数の VRF を作成し、各 VRF で同じ OSPFv2 インスタンスまたは複数の OSPFv3 インスタンスを使用することもできます。VRF に OSPFv2 インスタンスを割り当てることができます。

（注） インターフェイスの VRF を設定した後に、インターフェイスの他のすべてのパラメータを設定します。インターフェイスの VRF を設定すると、そのインターフェイスのすべての設定が削除されます。

始める前に

OSPF 機能を有効にしてあることを確認します（「OSPFv2 の有効化」の項を参照）。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **router ospf** *instance-tag*
4. **vrf** *vrf-name*
5. （任意） **maximum-paths** *path*
6. **interface** *interface-type slot/port*
7. **vrf member** *vrf-name*
8. **ip address** *ip-prefix/length*
9. **ip router ospf** *instance-tag* **area** *area-id*
10. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context RemoteOfficeVRF`
	`switch(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。

	コマンドまたはアクション	目的

Step 3	**router ospf** *instance-tag*
	例：
	`switch(config-vrf)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 4	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf RemoteOfficeVRF`
	`switch(config-router-vrf)#`
		VRF 設定モードを開始します。

Step 5	（任意） **maximum-paths** *path* 例： `switch(config-router-vrf)# maximum-paths 4`	この VRF のルート テーブル内の宛先への、同じ OSPFv2 パスの最大数を設定します。この機能は、ロード バランシングに使用されます。

Step 6	**interface** *interface-type slot/port*
	例：
	`switch(config-router-vrf)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 7	**vrf member** *vrf-name* 例： `switch(config-if)# vrf member RemoteOfficeVRF`	このインターフェイスを VRF に追加します。
Step 8	**ip address** *ip-prefix/length* 例： `switch(config-if)# ip address 192.0.2.1/16`	このインターフェイスの IP アドレスを設定します。このステップは、このインターフェイスを VRF に割り当てたあとに行う必要があります。
Step 9	**ip router ospf** *instance-tag* **area** *area-id* 例： `switch(config-if)# ip router ospf 201 area 0`	このインターフェイスを OSPFv2 インスタンスおよび設定エリアに割り当てます。
Step 10	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、VRF を作成して、その VRF にインターフェイスを追加する例を示します。

```
switch# configure terminal
switch(config)# vrf context NewVRF
switch(config)# router ospf 201
switch(config)# interface ethernet 1/2
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-if)# copy running-config startup-config
```

OSPFv2 設定の確認

OSPFv2 設定を表示するには、次のいずれかの作業を行います。

コマンド	目的
show ip ospf [*instance-tag*] [**vrf** *vrf-name*]	1 つ以上の OSPF ルーティング インスタンスに関する情報を表示します。出力には、次のエリア レベルのカウントが含まれます。 • このエリアのインターフェイス：このエリアに追加されたすべてのインターフェイスの数（設定されたインターフェイス）。 • アクティブ インターフェイス：ルータリンクステートおよび SPF（UP インターフェイス）にあると見なされるすべてのインターフェイスの数。 • パッシブ インターフェイス：OSPF パッシブと見なされるすべてのインターフェイスの数（隣接関係は形成されません）。 • ループバックインターフェイス：すべてのローカルループバックインターフェイスの数。
show ip ospf border-routers [**vrf** { *vrf-name* | **all** | **default** | **management** }]	OSPFv2 境界ルータ設定を表示します。
show ip ospf database [**vrf** { *vrf-name* | **all** | **default** | **management**}]	OSPFv2 リンクステートデータベースの要約を表示します。
show ip ospf interface *number* [**vrf** { *vrf-name* | **all** | **default** | **management** }]	OSPFv2-related インターフェイスの情報を表示します。
show ip ospf lsa-content-changed-list *neighbor-id interface - type number* [**vrf** { *vrf-name* | **all** | **default** | **management** }]	変更された OSPFv2 LSA を表示します。
show ip ospf neighbors [*neighbor-id*] [**detail**] [*interface - type number*] [**vrf** { *vrf-name* | **all** | **default** | **management** }] [**summary**]	OSPFv2 ネイバーの一覧を表示します。

コマンド	目的
show ip ospf request-list *neighbor-id interface - type number* [**vrf** {*vrf-name* | **all** | **default** | **management** }]	OSPFv2 リンクステート要求の一覧を表示します。
show ip ospf retransmission-list *neighbor-id interface - type number* [**vrf** { *vrf-name* | **all** | **default** | **management** }]	OSPFv2 リンクステート再送の一覧を表示します。
show ip ospf route [*ospf-route*] [**summary**] [**vrf** { *vrf-name* | **all** | **default** | **management** }]	内部 OSPFv2 ルートを表示します。
show ip ospf summary-address [**vrf** { *vrf-name* | **all** | **default** | **management** }]	OSPFv2 サマリ アドレスに関する情報を表示します。
show ip ospf virtual-links [**brief**] [**vrf** { *vrf-name* | **all** | **default** | **management** }]	OSPFv2 仮想リンクに関する情報を表示します。
show ip ospf vrf { *vrf-name* | **all** | **default** | **management** }	VRF ベースの OSPFv2 設定に関する情報を表示します。
show running-configuration ospf	現在実行中の OSPFv2 設定を表示します。

OSPFv2 のモニタリング

OSPFv2 統計情報を表示するには、次のコマンドを使用します。

コマンド	目的
show ip ospf policy statistics area *area-id* **filter list** {**in** | **out**} [**vrf** {*vrf-name* | **all** | **default** | **management**}]	エリアの OSPFv2 ルート ポリシー統計情報を表示します。
show ip policy statistics redestribute {**bgp** *id* | **direct** | **eigrp** *id* | **isis** *id* | **ospf** *id* | **rip** *id* | **static**} [**vrf** {*vrf-name* | **all** | **default** | **management**}]	OSPFv2 ルートポリシー統計情報を表示します。
show ip ospf statistics [**vrf** {*vrf-name* | **all** | **default** | **management**}]	OSPFv2 イベント カウンタを表示します。
show ip ospf traffic [*interface-type number*] [**vrf** {*vrf-name* | **all** | **default** | **management**}]	OSPFv2 パケット カウンタを表示します。

OSPFv2 の設定例

次に、OSPFv2 を設定する例を示します。

```
feature ospf
router ospf 201
```

```
 router-id 290.0.2.1
interface ethernet 1/2
 ip router ospf 201 area 0.0.0.10
 ip ospf authentication
 ip ospf authentication-key 0 mypass
```

OSPF RFC 互換モードの例

次に、RFC 1583 互換ルータと互換性を持つように OSPF を設定する例を示します。

（注） RFC1583 互換の OSPF のみを実行するルータに接続するすべての VRF で、RFC 1583 の互換性を設定する必要があります。

```
switch# configure terminal
switch(config)# feature ospf
switch(config)# router ospf Test1
switch(config-router)# rfc1583compatibility
switch(config-router)# vrf A
switch(config-router-vrf)# rfc1583compatibility
```

その他の参考資料

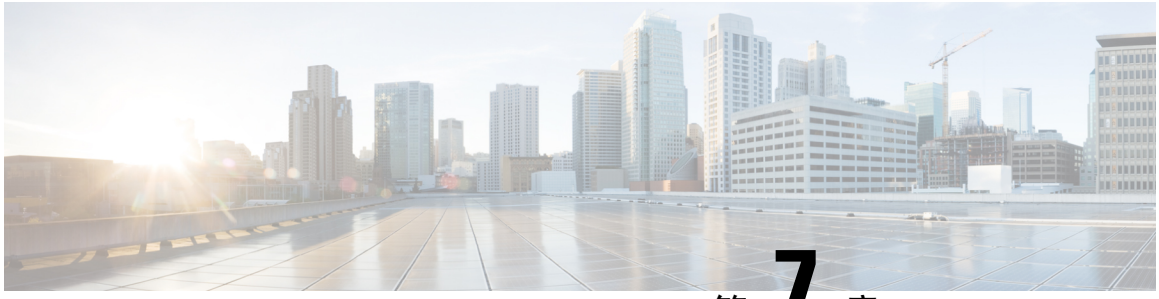
OSPF の実装に関する詳細情報については、次のページを参照してください。

OSPFv2 の関連資料

関連項目	マニュアル タイトル
キーチェーン	『Cisco Nexus 9000 Series NX-OS Security Configuration Guide』
IPv6 ネットワーク向け OSPFv3	OSPFv3 の設定 （169 ページ）
ルート マップ	Route Policy Manager の設定

MIB

MIB	MIB のリンク
OSPFv2 に関連する MIB	サポートされている MIB を検索およびダウンロードするには、次の URL にアクセスしてください。 ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html



第 7 章

OSPFv3 の設定

この章では、Cisco NX-OS デバイスで IPv6 ネットワーク用の Open Shortest Path First version 3（OSPFv3）を設定する方法について説明します。

この章は、次の項で構成されています。

- OSPFv3 について （169 ページ）
- マルチエリア隣接関係（Multi-Area Adjacency） （177 ページ）
- OSPFv3 と IPv6 ユニキャスト RIB （177 ページ）
- アドレス ファミリのサポート （177 ページ）
- 認証および暗号化 （178 ページ）
- 高度な機能 （178 ページ）
- OSPFv3 の前提条件 （183 ページ）
- OSPFv3 の注意事項および制約事項 （184 ページ）
- デフォルト設定 （186 ページ）
- 基本的なOSPFv3の設定 （186 ページ）
- 高度なOSPFv3の設定 （193 ページ）
- 暗号化および認証の構成 （219 ページ）
- OSPFv3 の設定の確認 （232 ページ）
- OSPFv3のモニタリング （234 ページ）
- OSPFv3 の設定例 （234 ページ）
- 関連項目 （235 ページ）
- その他の参考資料 （235 ページ）

OSPFv3 について

OSPFv3 は、IETF リンクステート プロトコル（概要 （3 ページ）の項を参照）です。OSPFv3 ルータは、hello パケットと呼ばれる特別なメッセージを各 OSPF 対応インターフェイスに送信し、他の OSPFv3 隣接ルータを探索します。ネイバー ルータが発見されると、この 2 台のルータは hello パケットの情報を比較して、両者の設定に互換性のあるかどうかを判定します。これらのネイバールータは隣接を確立しようとします。つまり、両者のリンクステートデータベースを同期させて、確実に同じ OSPFv3 ルーティング情報を持つようにします。隣接ルータは、各リンクの

稼働状態に関する情報、リンクのコスト、およびその他のあらゆるネイバー情報を含むリンクステートアドバタイズメント（LSA）を共有します。これらのルータはその後、受信したLSAをすべてのOSPFイネーブルインターフェイスにフラッディングします。これにより、すべてのOSPFv3ルータのリンクステート データベースが最終的に同じになります。すべての OSPFv3 ルータのリンクステート データベースが同じになると、ネットワークは収束します（「コンバージェンス」を参照）。その後、各ルータは、ダイクストラの最短パス優先（SPF）アルゴリズムを使用して、自身のルート テーブルを構築します。

OSPFv3 ネットワークは、複数のエリアに分割できます。ルータは、ほとんどの LSA を 1 つのエリア内だけに送信するため、OSPF 対応ルータの CPU とメモリの要件が緩やかになります。

OSPFv3 はIPv6をサポートしています。IPv4 向けの OSPF の詳細については、OSPFv2 の設定（113 ページ）を参照してください。

OSPFv3 と OSPFv2 の比較

OSPFv3 プロトコルの大半は OSPFv2 と同じです。OSPFv3 は RFC 2740 に記載されています。

OSPFv3 プロトコルと OSPFv2 プロトコルの重要な相違点は、次のとおりです。

- OSPFv2 を拡張した OSPFv3 では、IPv6 ルーティング プレフィックスとサイズの大きい IPv6 アドレスのサポートを提供しています。
- OSPFv3 の LSA は、アドレスとマスクではなく、プレフィックスとプレフィックス長として表現されます。
- ルータ ID とエリア ID は 32 ビット数で、IPv6 アドレスとは無関係です。
- OSPFv3 では、ネイバー探索およびその他の機能にリンクローカル IPv6 アドレスを使用します。
- OSPFv3 は、IPv6 認証トレーラ（RFC 6506）または IPSec（RFC 4552）を使用できます。ただし、Cisco NX-OS は RFC 6506 をサポートしていません。
- OSPFv3 では、LSA タイプが再定義されています。

Hello パケット

OSPFv3 ルータは、すべての OSPF イネーブル インターフェイスに hello パケットを定期的に送信します。ルータがこの hello パケットを送信する頻度は、インターフェイスごとに設定された hello 間隔により決定されます。OSPFv3 は、hello パケットを使用して、次のタスクを実行します。

- ネイバー探索
- キープアライブ
- 双方向通信
- 指定ルータの選定（「指定ルータ」セクションを参照してください）

helloパケットには、リンクのOSPFv3コスト割り当て、hello間隔、送信元ルータのオプション機能など、送信元のOSPFv3インターフェイスとルータに関する情報が含まれます。これらのhelloパケットを受信するOSPFv3インターフェイスは、設定に受信インターフェイスの設定との互換性があるかどうかを判定します。互換性のあるインターフェイスはネイバーと見なされ、ネイバーテーブルに追加されます（「ネイバー」の項を参照してください）。

helloパケットには、送信元インターフェイスが通信したルータのルータIDのリストも含まれます。受信インターフェイスが、このリストで自身のIDを見つけた場合は、2つのインターフェイス間で双方向通信が確立されます。

OSPFv3は、helloパケットをキープアライブ メッセージとして使用して、ネイバーが通信を継続中であるかどうかを判定します。ルータが設定されたデッド間隔（通常はhello間隔の倍数）でhelloパケットを受信しない場合、そのネイバーはローカル ネイバーテーブルから削除されます。

ネイバー情報

ネイバーであると見なされるようにするには、リモートインターフェイスと互換性があるようにOSPFv3インターフェイスを設定しておく必要があります。この2つのOSPFv3インターフェイスで、次の基準が一致している必要があります。

- hello間隔
- デッド間隔
- エリアID（「エリア」の項を参照）
- オプション機能

一致する場合は、次の情報がネイバー テーブルに入力されます。

- ネイバーID：ネイバー ルータのルータID
- 優先度：ネイバー ルータの優先度。プライオリティは、指定ルータの選定（「指定ルータ」を参照）に使用されます。
- 状態：ネイバーから通信があったか、双方向通信の確立処理中であるか、リンクステート情報を共有しているか、または完全な隣接関係が確立されたかを示します。
- デッド タイム：このネイバーから最後のhelloパケットを受信したあとに経過した時間を示します。
- リンクローカルIPv6アドレス：ネイバーのリンクローカルIPv6アドレス
- 指定ルータ：ネイバーが指定ルータ、またはバックアップ指定ルータとして宣言されたかどうかを示します（「指定ルータ」の項を参照）。
- ローカルインターフェイス：このネイバーのhelloパケットを受信したローカルインターフェイス。

最初のhelloパケットが新規ネイバーから受信されると、そのネイバーは、初期化状態のネイバーテーブルに入力されます。いったん双方向通信が確立されると、ネイバー状態は双方向となりま

す。2 つのインターフェイスが互いのリンクステート データベースを交換するため、次に ExStart および交換状態となります。これらがすべて完了すると、ネイバーは完全な状態へと移行し、これが完全な隣接関係となります。ネイバーは、デッド間隔でhelloパケットをまったく送信しない場合は、ダウン状態に移行し、隣接とは見なされなくなります。

隣接関係

すべてのネイバーが隣接関係を確立するわけではありません。ネットワーク タイプと確立された指定ルータに応じて、完全な隣接関係を確立して、すべてのネイバーと LSA を共有するものと、そうでないものがあります。詳細については、「指定されたルータ」の項を参照してください。

隣接関係は、OSPFv3 のデータベース説明（DD）パケット、リンク状態要求（LSR）パケット、およびリンク状態更新（LSU）パケットを使用して確立されます。データベース説明パケットには、ネイバーのリンクステート データベースからの LSA ヘッダーが含まれます（「リンク状態データベース 」の項を参照）。ローカル ルータは、これらのヘッダーを自身のリンクステート データベースと比較して、新規の LSA か、更新された LSA かを判定します。ローカル ルータは、新規または更新の情報を必要とする各 LSA について、リンク状態要求（LSR）パケットを送信します。ネイバーは LSU パケットで応答します。このパケット交換は、両方のルータのリンクステート情報が同じになるまで継続します。

指定ルータ

複数のルータを含むネットワークは、OSPFv3 特有の状況です。すべてのルータがネットワークで LSA をフラッディングした場合は、同じリンクステート情報が複数の送信元から送信されます。ネットワークのタイプによっては、OSPFv3 は指定ルータ（DR）という 1 台のルータを使用して LSA のフラッディングを制御し、OSPFv3 の残りの部分に対してネットワークを代表する役割をさせる場合があります（「エリア」の項を参照）。DR がダウンした場合、OSPFv3 はバックアップ指定ルータ（BDR）を選択します。DR がダウンすると、OSPFv3 はこの BDR を使用します。

ネットワーク タイプは次のとおりです。

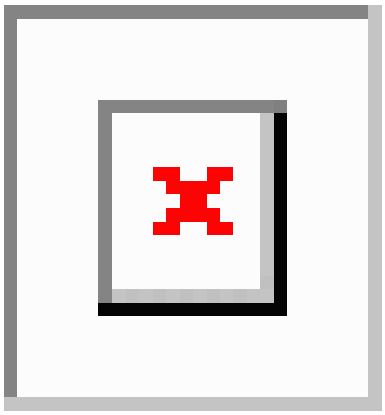
- ポイントツーポイント： 2 台のルータ間にのみ存在するネットワーク。ポイントツーポイント ネットワーク上の全ネイバーは隣接関係を確立し、DR は存在しません。
- ブロードキャスト：ブロードキャスト トラフィックが可能なイーサネットなどの共有メディア上で通信できる複数のルータを持つネットワーク。OSPFv3 ルータは DR および BDR を確立し、これらにより、ネットワーク上の LSA フラッディングを制御します。OSPFv3 は、よく知られている IPv6 マルチキャスト アドレス FF02::5 および MAC アドレス 0100.5300.0005 を使用して、ネイバーと通信します。

DR と BDR は、hello パケット内の情報に基づいて選択されます。インターフェイスはhelloパケットの送信時に、どれが DR および BDR かわかっている場合は、優先フィールドと、DR および BDR フィールドを設定します。ルータは、hello パケットの DR および BDR フィールドで宣言されたルータと優先フィールドに基づいて、選定手順を実行します。最終的に OSPFv3 は、最も大きいルータ ID を DR および BDR として選択します。

他のルータはすべて DR および BDR と隣接関係を確立し、IPv6 マルチキャスト アドレス FF02::6 を使用して、LSA 更新情報を DR と BDR に送信します。次の図は、すべてのルータと DR との隣接関係を示しています。

DR は、ルータ インターフェイスに基づいています。1 つのネットワークの DR であるルータは、別のインターフェイス上の他のネットワークの DR となることはできません。

図 *22*：マルチアクセス ネットワークの *DR*



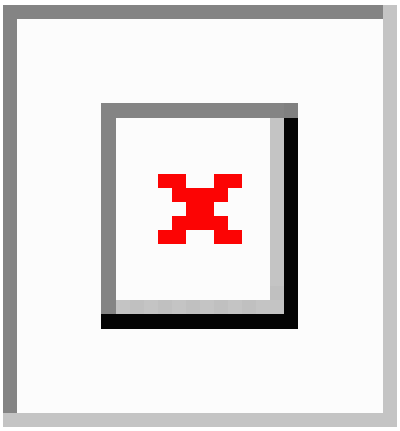
エリア

OSPFv3 ネットワークを複数のエリアに分割すると、ルータに要求される OSPFv3 の CPU とメモリに関する要件を制限できます。エリアとは、ルータの論理的な区分で、OSPFv3 ドメイン内にリンクして別のサブドメインを作成します。LSA フラッディングはエリア内でのみ発生し、リンク ステート データベースはエリア内のリンクにのみ制限されます。定義されたエリア内のインターフェイスには、エリア ID を割り当てることができます。エリア ID は、10.2.3.1 などの、数字またはドット付き 10 進表記で表現される 32 ビット値です。

Cisco NX-OS は常にドット付き 10 進表記でエリアを表示します。

OSPFv3 ネットワーク内に複数のエリアを定義する場合は、0 という予約されたエリア ID を持つバックボーン エリアも定義する必要があります。エリアが複数ある場合は、1 台以上のルータがエリア境界ルータ（ABR）となります。ABR は、バックボーン エリアと他の 1 つ以上の定義済みエリアの両方に接続します。

図 *23 : OSPFv3* エリア



ABR には、接続するエリアごとに個別のリンクステート データベースがあります。ABR は、接続したエリアの 1 つからバックボーンエリアにエリア間プレフィックス（タイプ 3）LSA（「ルート集約」セクションを参照）を送信します。バックボーン エリアは、1 つのエリアに関する集約情報を別のエリアに送信します。図に、エリア 0 が、エリア 5 に関する集約情報をエリア 3 に送信しています。

OSPFv3 では、自律システム境界ルータ（ASBR）という、もう 1 つのルータ タイプも定義されています。このルータは、OSPFv3 エリアを別の自律システム（AS）に接続します。自律システムとは、単一の技術的管理エンティティにより制御されるネットワークです。OSPFv3 は、そのルーティング情報を別の自律システムに再配布したり、再配布されたルートを別の自律システムから受信したりできます。詳細については、「詳細な機能」のセクションを参照してください。

リンクステート アドバタイズメント

OSPFv3 はリンクステート アドバタイズメント（LSA）を使用して、固有のルーティング テーブルを構築します。

リンクステート アドバタイズメント タイプ

OSPFv3 はリンクステート アドバタイズメント（LSA）を使用して、固有のルーティング テーブルを構築します。

次の表に、Cisco NX-OS でサポートされる LSA タイプを示します。

タイプ	名前	説明
1	ルータ LSA	すべてのルータが送信する LSA。この LSA には、すべてのリンクの状態とコストが含まれますが、プレフィックス情報は含まれません。ルータ LSA は SPF 再計算をトリガーします。ルータ LSA はローカル OSPFv3 エリアにフラッディングされます。

タイプ	名前	説明
2	ネットワーク LSA	DR が送信する LSA。この LSA には、マルチアクセス ネットワーク内のすべてのルータの一覧が含まれますが、プレフィックス情報は含まれません。ネットワーク LSA は SPF 再計算をトリガーします。「指定ルータ」のセクションを参照してください。
3	エリア間プレフィックス LSA	ABR が、ローカル エリア内の宛先ごとに外部エリアに送信する LSA。この LSA には、境界ルータからローカルの宛先へのリンク コストが含まれます。「エリア」のセクションを参照してください。
4	エリア間ルータ LSA	エリア境界ルータが外部エリアに送信する LSA。この LSA は、リンク コストを ASBR のみにアドバタイズします。「エリア」の項を参照してください。
5	AS 外部 LSA	ASBR が生成する LSA。この LSA には、外部自律システム宛先へのリンク コストが含まれます。AS 外部 LSA は、自律システム全体にわたってフラッディングされます。「エリア」の項を参照してください。
7	タイプ 7 LSA	ASBR が NSSA 内で生成する LSA。この LSA には、外部自律システム宛先へのリンク コストが含まれます。タイプ 7 LSA は、ローカル NSSA 内のみでフラッディングされます。「エリア」の項を参照してください。
8	リンク LSA	各ルータが、リンクローカル フラッディング スコープを使用して送信する LSA。（「フラッディングと LSA グループ ペーシング」の項を参照）。この LSA には、このリンクのリンクローカル アドレスと IPv6 アドレスが含まれます。
9	エリア内プレフィックス LSA	すべてのルータが送信する LSA。この LSA には、プレフィックスまたはリンク状態へのあらゆる変更が含まれます。エリア内プレフィックス LSA はローカル OSPFv3 エリアにフラッディングされます。この LSA は SPF 再計算をトリガーしません。
11	Grace LSA	再起動されるルータが、リンクローカル フラッディング スコープを使用して送信する LSA。この LSA は、OSPFv3 のグレースフル リスタートに使用されます。「ハイ アベイラビリティおよびグレースフル リスタート」を参照してください。

リンク コスト

各 OSPFv3 インターフェイスは、リンク コストを割り当てられています。このコストは任意の数字です。デフォルトでは、Cisco NX-OS が、設定された参照帯域幅をインターフェイス帯域幅で割った値をコストとして割り当てます。デフォルトでは、参照帯域幅は 40 Gbps です。リンク コストは各リンクに対して、LSA 更新情報で伝えられます。

フラッディングと LSA グループ ペーシング

OSPFv3 は、LSA のタイプに応じて、ネットワークのさまざまなセクションに LSA の更新をフラッディングします。OSPFv3 は、次のフラッディング スコープを使用します

- リンク ローカル：LSA は、ローカル リンク上でのみフラッディングされます。リンク LSA および猶予 LSA に使用されます。
- エリアローカル：LSA は、単一の OSPF エリア全体にのみフラッディングされます。ルータ LSA、ネットワーク LSA、エリア間プレフィックス LSAs、エリア間ルータ LSA、およびエリア内プレフィックス LSA に使用されます。
- AS スコープ：LSA は、ルーティング ドメイン全体にフラッディングされます。AS スコープは AS 外部 LSA に使用されます。

LSA フラッディングにより、ネットワーク内のすべてのルータが同じルーティング情報を持つことが保証されます。LSA フラッディングは、OSPFv3 エリアの設定により異なります（「エリア」の項を参照）。LSA は、リンクステート リフレッシュ時間に基づいて（デフォルトでは 30 分ごとに）フラッディングされます。各 LSA には、リンクステート リフレッシュ時間が設定されています。

ネットワークの LSA 更新情報のフラッディング レートは、LSA グループ ペーシング機能を使用して制御できます。LSA グループ ペーシングにより、CPU またはバッファの使用率を低下させることができます。この機能により、同様のリンクステート リフレッシュ時間を持つ LSA がグループ化されるため、OSPFv3 で、複数の LSA を 1 つの OSPFv3 更新メッセージにまとめることが可能となります。

デフォルトでは、相互のリンクステート リフレッシュ時間が 10 秒以内の LSA が、同じグループに入れられます。この値は、大規模なリンクステート データベースでは低く、小規模のデータベースでは高くして、ネットワーク上の OSPFv3 負荷を最適化する必要があります。

リンクステート データベース

各ルータは、OSPFv3 ネットワーク用のリンクステート データベースを保持しています。このデータベースには、収集されたすべての LSA が含まれ、ネットワークを通過するすべてのルートに関する情報が格納されます。OSPFv3 は、この情報を使用して、各宛先への最適なパスを計算し、この最適なパスをルーティング テーブルに入力します。

MaxAge と呼ばれる設定済みの時間間隔で受信された LSA 更新情報がまったくない場合は、リンクステート データベースから LSA が削除されます。ルータは、LSA を 30 分ごとに繰り返してフラッディングし、正確なリンクステート情報が期限切れで削除されるのを防ぎます。Cisco NX-OS は、LSA グルーピング機能をサポートし、同時にすべての LSA が更新されないようにします。詳細については、「フラッディングと LSA グループ ペーシング 」のセクションを参照してください。

マルチエリア隣接関係（Multi-Area Adjacency）

OSPFv3 マルチエリア隣接関係により、複数のエリアにあるプライマリ インターフェイス上にリンクを設定できます。このリンクは、それらのエリア内の優先されるエリア内リンクになります。マルチエリア隣接関係では、OSPFv3エリアにポイントツーポイントの番号なしリンクを確立し、そのエリアにトポロジパスを提供します。プライマリ隣接関係はリンクを使用して、ネイバーステートが full の場合に、ルータ LSA で対応するエリアの番号なしポイントツーポイント リンクをアドバタイズします。

マルチエリア インターフェイスは、OSPF の既存のプライマリ インターフェイス上の論理構成体として存在しますが、プライマリ インターフェイス上のネイバー ステートは、マルチエリア インターフェイスと無関係です。マルチエリアインターフェイスはネイバールータ上の対応するマルチエリア インターフェイスとの隣接関係を確立します。詳細については、「マルチエリア隣接関係の設定」の項を参照してください。

OSPFv3 と IPv6 ユニキャスト RIB

OSPFv3 は、リンクステート データベースでダイクストラの SPF アルゴリズムを実行します。このアルゴリズムにより、パス上の各リンクのリンク コストの合計に基づいて、各宛先への最適なパスが選択されます。選択された各宛先への最短パスがOSPFv3ルートテーブルに入力されます。OSPFv3 ネットワークが収束すると、このルート テーブルは IPv6 ユニキャスト ルーティング情報ベース（RIB）にデータを提供します。OSPFv3 は IPv6 ユニキャスト RIB と通信し、次の動作を行います。

- ルートの追加または削除
- 他のプロトコルからのルートの再配布への対応
- 変更されていない OSPFv3 ルートの削除およびスタブ ルータ アドバタイズメントを行うためのコンバージェンス更新情報を提供します（「複数の OSPFv3 インスタンス（Multiple OSPFv3 Instances）」を参照）。

さらに OSPFv3 は、変更済みダイクストラ アルゴリズムを実行して、エリア間プレフィックス、エリア間ルータ、AS 外部、タイプ 7、およびエリア内プレフィックス（タイプ 3、4、5、7、8）の各 LSA の変更の高速再計算を行います。

アドレス ファミリのサポート

Cisco NX-OS は、ユニキャスト IPv6 やマルチキャスト IPv6 などの複数のアドレス ファミリをサポートしています。アドレス ファミリに特有の OSPFv3 機能は、次のとおりです。

- デフォルト ルート
- ルート集約

- ルートの再配布
- 境界ルータのフィルタ リスト
- SPF 最適化

これらの機能の設定時に IPv6 ユニキャスト アドレス ファミリ コンフィギュレーション モードを開始するには、**address-family ipv6 unicast** コマンドを使用します。

認証および暗号化

OSPFv3 メッセージに認証を設定して、ネットワークでの不正な、または無効なルーティング更新を防止できます。

RFC 4552 は、IPv6 認証ヘッダー (AH) またはカプセル化セキュリティ ペイロード (ESP) 拡張ヘッダーを使用して、OSPFv3 への認証を提供します。Cisco NX-OS は、IPv6 AH ヘッダーを使用して OSPFv3 パケットを認証することにより、RFC 4552 をサポートします。

Cisco NX-OS は、IP セキュリティ（IPSec）認証方式と、メッセージ ダイジェスト 5（MD5）またはセキュア ハッシュ アルゴリズム 1（SHA1）アルゴリズムをサポートして、OSPFv3 パケットを認証します。OSPFv3 IPSec 認証は、コマンドを使用しする静的キーのみをサポートします。

Cisco NX-OS は、OSPFv3 メッセージの暗号化と認証の両方に IPSec ESP 方式もサポートしています。暗号化は、ESP 暗号化の AES または 3DES アルゴリズムと、ESP 認証の SHA-1 または NULL をサポートします。

Cisco NX-OS リリース 10.4(1)F 以降、Cisco NX-OS は、キーチェーン オプションを使用した暗号化または認証アルゴリズムとキーの構成をサポートしています。

IPSec 暗号化または認証は、OSPFv3 プロセス、エリア、インターフェイス、あるいはその両方に対して構成可能です。認証設定は、プロセスからエリア、インターフェイス レベルに継承されます。認証が 3 つのレベルすべてで構成されている場合、インターフェイス構成がプロセスおよびエリア構成よりも優先され、エリア構成はプロセス レベルよりも優先されます。

高度な機能

Cisco NX-OS は、ネットワークでの OSPFv3 の可用性やスケーラビリティを向上させる高度な OSPFv3 機能をサポートしています。

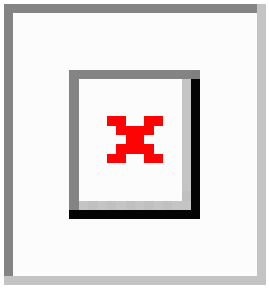
スタブ エリア

エリアをスタブ エリアにすると、エリアでフラッディングされる外部ルーティング情報の量を制限できます。スタブエリアとは、AS 外部（タイプ 5）LSA（リンクステートアドバタイズメント（174 ページ）の項を参照）が許可されないエリアです。これらの LSA は通常、外部ルーティング情報を伝播するためにローカル自律システム全体でフラッディングされます。スタブ エリアには、次の要件があります。

- スタブ エリア内のすべてのルータはスタブ ルータです。「スタブ ルーティング」の項を参照してください。
- スタブ エリアには ASBR ルータは存在しません。
- スタブ エリアには仮想リンクを設定できません。

次の図に示すOSPFv3 自律システムでは、エリア 0.0.0.10 内のルータはすべて、外部自律システムに到達するために ABR を通過しなければなりません。エリア 0.0.0.10 は、スタブ エリアとして設定できます。

図 *24 :* スタブ エリア



スタブエリアは、外部自律システムへのバックボーンエリアを通過する必要のあるすべてのトラフィックにデフォルト ルートを使用します。デフォルト ルートは、プレフィックス長が IPv6 向けに 0 に設定されたエリア間プレフィックス LSA です。

Not-So-Stubby Area

Not-So-Stubby Area（NSSA）は、スタブ エリアに似ていますが、NSSA では、再配布を使用して NSSA 内で自律システム外部ルートをインポートできる点が異なります。NSSA ASBR はこれらのルートを再配布し、タイプ 7 LSA を生成して NSSA 全体にフラッディングします。または、このタイプ 7 LSA を AS 外部（タイプ 5）LSA に変換するよう、NSSA を他のエリアに接続する ABR を設定することができます。こうすると、ABR は、これらの AS 外部 LSA を OSPFv3 自律システム全体にフラッディングします。変換中は集約とフィルタリングがサポートされます。タイプ 7 LSA の詳細については、リンクステート アドバタイズメント（174 ページ）の項を参照してください。

たとえば、OSPFv3 を使用する中央サイトを、異なるルーティング プロトコルを使用するリモート サイトに接続するときに NSSA を使用すると、管理作業を簡素化できます。NSSA を使用する前は、企業サイトの境界ルータとリモート ルータの間の接続を OSPFv3 スタブ エリアとして実行できませんでした。これは、リモートサイトへのルートはスタブエリア内に再配布できないためです。NSSA が実装されたことで、企業ルータとリモートルータ間のエリアを NSSA として定義することにより、NSSA で OSPFv3 を拡張してリモート接続をカバーできます。（「NSSA の設定」の項を参照）。

バックボーン エリア 0 を NSSA にできません

（注） Cisco NX-OS リリース 9.3(1) 以降、OSPF は RFC 3101 セクション 2.5(3) に準拠するようになりました。Not-so-Stubby Area に接続されたエリア境界ルータが P ビット クリアのデフォルト ルート LSA を受信した場合は、無視されます。OSPF は、これらの条件下で以前にデフォルト ルートを追加していました。

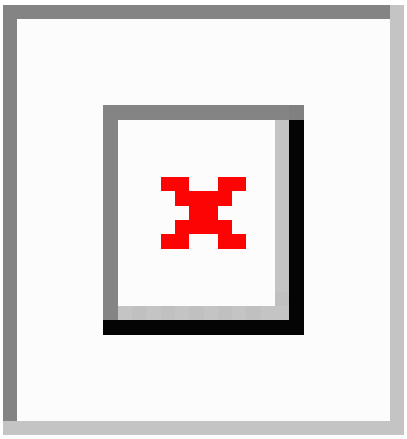
すでに RFC 非準拠の動作を使用するようにネットワークを設計しており、デフォルト ルートが NSSA ABR に追加されると想定している場合は、Cisco NX-OS リリース 9.3(1) 以降にアップグレードするときに動作が変更されます。

古い動作を続行する場合は、**default-route nssa-abr pbit-clear** コマンドで有効にすることができます。このコマンドは、Cisco NX-OS Release 9.3(1) で実装されました。

仮想リンク

仮想リンクを使用すると、物理的に直接接続できない場合に、OSPFv3 エリア ABR をバックボーン エリア ABR に接続できます。図には、エリア 3 をエリア 5 経由でバックボーン エリアに接続する仮想リンクを示します。

図 *25 :* 仮想リンク



また、仮想リンクを使用して、分割エリアから一時的に回復できます。分割エリアは、エリア内のリンクがダウンしたために隔離された一部のエリアで、ここからはバックボーン エリアへの代表 ABR に到達できません。

ルートの再配布

OSPFv3 は、ルート再配布を使用して、他のルーティング プロトコルからルートを学習できます。「ルートの再配布の概要 （8 ページ）」の項を参照してください。リンク コストをこれらの再配布されたルートに割り当てるか、またはデフォルト リンク コストを再配布されたすべてのに割り当てるよう、OSPFv3 を設定します。

ルート再配布では、ルート マップを使用して、再配布する外部ルートを管理します。再配布を指定したルートマップを設定して、どのルートがOSPFv3に渡されるかを制御する必要があります。ルート マップを使用すると、宛先、送信元プロトコル、ルート タイプ、ルート タグなどの属性に基づいて、ルートをフィルタリングできます。ルート マップを使用して、これらの外部ルートがローカルOSPFv3 ASでアドバタイズされる前にAS外部（タイプ5）LSAおよびNSSA外部（タイプ7）LSA のパラメータを変更できます。詳細については、Route Policy Manager の設定 （587 ページ） を参照してください。

ルート集約

OSPFv3は学習したすべてのルートをあらゆるOSPF対応ルータと共有するので、ルート集約を使用して、それぞれの OSPF 対応ルータにフラッディングされる固有のルートの数を削減した方がよい場合もあります。ルート集約により、より具体的な複数のアドレスが、すべての具体的なアドレスを表す1つのアドレスに置き換えられるため、ルート テーブルが簡素化されます。たとえば、2010:11:22:0:1000::1 と 2010:11:22:0:2000:679:1 を 1 つの集約アドレス 2010:11:22::/32 に置き換えることができます。

一般的には、エリア境界ルータ（ABR）の境界ごとに集約します。集約は 2 つのエリアの間でも設定できますが、バックボーンの方向に集約する方が適切です。こうすると、バックボーンがすべての集約アドレスを受信し、すでに集約されているそれらのアドレスを他のエリアに投入できるためです。集約には、次の 2 タイプがあります。

- エリア間ルート集約
- 外部ルート集約

エリア間ルート集約はABR上で設定し、自律システム内のエリア間のルートを集約します。集約の利点を生かすには、これらのアドレスを 1 つの範囲内にまとめることができるように、連続するネットワーク番号をエリア内で割り当てます。

外部ルート集約は、ルート再配布を使用して OSPFv3 に投入される外部ルートに特有のルート集約です。集約する外部の範囲が連続していることを確認する必要があります。異なる 2 台のルータからの重複範囲を集約すると、誤った宛先にパケットが送信される原因となる場合があります。外部ルート集約は、ルートを OSPF に再配布している ASBR で設定してください。

集約アドレスの設定時に Cisco NX-OS は、ルーティング ブラック ホールおよびルート ループを防ぐために、集約アドレスの廃棄ルートを自動的に設定します。

高可用性およびグレースフル リスタート

Cisco NX-OS は、マルチレベルのハイ アベイラビリティ アーキテクチャを提供します。OSPFv3 は、ステートフル リスタートをサポートしています。これは、ノンストップルーティング（NSR）とも呼ばれます。OSPFv3 で問題が発生した場合は、以前の実行時状態からの再起動を試みます。この場合、ネイバーはいずれのネイバー イベントも登録しません。最初の再起動が正常ではなく、別の問題が発生した場合、OSPFv3 はグレースフル リスタートを試みます。

グレースフル リスタート、つまり、Nonstop Forwarding（NSF）では、処理の再起動中も OSPFv3 がデータ転送パス上に存在し続けます。OSPFv3 はグレースフル リスタートの実行が必要になる

と、リンクローカル猶予（タイプ 11）LSA を送信します。この再起動中の OSPFv3 プラットフォームは NSF 対応と呼ばれます。

猶予 LSA には猶予期間が含まれます。猶予期間とは、ネイバー OSPFv3 インターフェイスは再起動中の OSPFv3 インターフェイスからの LSA を待つよう指定された時間です（通常、OSPFv3は隣接関係を切断し、ダウン状態または再起動中のOSPFv3インターフェイスからのすべてのLSAを廃棄します）。参加するネイバーは、NSF ヘルパーと呼ばれ、再起動中の OSPFv3 インターフェイスから発信されたすべての LSA を、インターフェイスがまだ隣接しているかのように保持します。

再起動中の OSPFv3 インターフェイスが稼働を再開すると、ネイバーを再探索して隣接関係を確立し、LSA 更新情報の送信を再開します。この時点で、NSF ヘルパーは、グレースフル リスタートが完了したと認識します。

ステートフル リスタートは次のシナリオで使用されます。

- プロセスでの問題発生後の最初の回復試行
- **system switchover** を使用したユーザ開始スイッチオーバー command

グレースフル リスタートは次のシナリオで使用されます。

- プロセスでの問題発生後の 2 回目の回復試行（4 分以内）
- **restart ospfv3** を使用したプロセスの手動再起動 command
- アクティブ スーパーバイザの削除
- **reload module** *active-sup* コマンド

複数の OSPFv3 インスタンス

Cisco NX-OS は、OSPFv3 プロトコルの複数インスタンスをサポートしています。デフォルトでは、すべてのインスタンスが同じシステム ルータ ID を使用します。複数のインスタンスが同じ OSPFv3 自律システムにある場合は、各インスタンスのルータ ID を手動で設定する必要があります。サポートされる OSPFv3 インスタンスの数については、『Cisco Nexus 9000 Series NX-OS Verified Scalability Guide』を参照してください。

OSPFv3 ヘッダーには、特定の OSPFv3 インスタンスの OSPFv3 パケットを識別するためのインスタンス ID フィールドが含まれます。この OSPv3 インスタンスを割り当てることができます。インターフェイスは、パケット ヘッダーの OSPFv3 インスタンス ID が一致しない OSPFv3 パケットをすべてドロップします。

Cisco NX-OS では、インターフェイス上に 1 つの OSPFv3 インスタンスのみが許可されます。

SPF 最適化

Cisco NX-OS は、次の方法で SPF アルゴリズムを最適化します。

- ネットワーク（タイプ 2）LSA、エリア間プレフィックス（タイプ 3）LSA、および AS 外部（タイプ 5）LSA 用部分 SPF：これらの LSA のいずれかが変更されると、Cisco NX-OS は、全体的な SPF 計算ではなく、高速部分計算を実行します。
- SPF タイマー：さまざまなタイマーを設定して、SPF 計算を制御できます。これらのタイマーには、後続の SPF 計算の幾何バックオフが含まれます。幾何バックオフにより、複数の SPF 計算による CPU 負荷が制限されます。

BFD

この機能では、IPv6 用の双方向フォワーディング検出（BFD）をサポートします。BFD は、転送パスの障害を高速で検出することを目的にした検出プロトコルです。BFD は 2 台の隣接デバイス間のサブセカンド障害を検出し、BFD の負荷の一部を、サポートされるモジュール上のデータプレーンに分散できるため、プロトコル hello メッセージよりも CPU を使いません。詳細については、『Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide』を参照してください。

仮想化のサポート

Cisco NX-OS は、OSPFv3 の複数のプロセスインスタンスをサポートします。各 OSPFv3 インスタンスは、システム制限まで、複数の仮想ルーティングおよび転送（VRF）インスタンスをサポートできます。サポートされる OSPFv3 インスタンスの数については、『Cisco Nexus 9000 Series NX-OS Verified Scalability Guide』を参照してください。

OSPFv3 の前提条件

OSPFv3 の前提条件は次のとおりです。

- OSPFv3 を設定するための、ルーティングの基礎に関する詳しい知識がある。
- スイッチにログオンしている。
- リモート OSPFv3 ネイバーと通信可能な 1 つ以上の IPv6 用インターフェイスが設定されている。
- Enterprise Services ライセンスがインストールされている。
- OSPFv3 ネットワーク戦略と、ネットワークのプランニングが完成している。たとえば、複数のエリアが必要かどうかを決定します。
- OSPF 機能を有効していること（「OSPFv3 の有効化 」の項を参照）。
- IPv6 アドレス指定および基本設定に関する詳しい知識がある。IPv6 ルーティングおよびアドレス指定の詳細については、IPv6 の設定 （59 ページ）を参照してください。

OSPFv3 の注意事項および制約事項

OSPFv3 設定時の注意事項および制約事項は、次のとおりです。

- リロード時の OSPFv2 の **graceful-restart planned-only** コマンドは、**graceful-restart** コマンドに変換されます。

  これは機能に影響を与えません。**graceful-restart planned-only** が設定にない場合、この問題はそのデバイスには適用されません。

  これは、Cisco NX-OS リリースが 9.3(2) で、CSCvs57583 がリリースに含まれていない場合に発生します。回避策は、**graceful-restart** コマンドを設定解除し、古いコマンドを再設定することです。

- プレフィックス リスト内の名前は、大文字と小文字が区別されません。一意の名前を使用することを推奨します。大文字と小文字を変更して同じ名前を使用しないでください。たとえば、CTCPrimaryNetworks と CtcPrimaryNetworks は 2 つの異なるエントリではありません。

- **no graceful-restart planned only** コマンドを入力すると、グレースフル リスタートは無効になります。

- Cisco NX-OS は、ユーザがエリアを 10 進表記で入力するか、ドット付き 10 進表記で入力するかに関係なく、ドット付き 10 進表記でエリアを表示します。

- 仮想ポート チャネル（vPC）環境で OSPFv3 を設定する場合は、コア スイッチ上のルータ コンフィギュレーション モードで次のタイマー コマンドを使用することにより、vPC ピアリンクがシャットダウンしたときに OSPF の高速コンバージェンスを実現します。

  ```
  switch(config-router)# timers throttle spf 1 50 50
  switch(config-router)# timers lsa-arrival 10
  ```

- スケール シナリオでは、インターフェイスと OSPF プロセスのリンク ステート アドバタイズメントの数が大きい場合、OSPF MIB オブジェクトの SNMP エージェントのタイムアウト値が小さい SNMP ウォークは、タイムアウトになると予想されます。OSPF MIB オブジェクトのポーリング中に問い合わせる SNMP エージェントのタイムアウトを確認する場合は、ポーリングする SNMP エージェントのタイムアウト値を増加してください。

- アドミニストレーティブ ディスタンス機能には、次のガイドラインと制限事項が適用されます。

  - OSPF ルートに複数の等コスト パスがある場合、アドミニストレーティブ ディスタンスを設定しても **match ip route-source** コマンドに対しては決定性を持ちません。 コマンドを使用する必要があります。

  - OSPFv3ルートのルートソースを照合するには、 **match ip route-source** を設定します。 次は古い構文です: **match ipv6 route-source** OSPFv3 のルートソースとルータ ID が IPv4 アドレスであるためです。

  - アドミニストレーティブ ディスタンスの設定は、**match route-type**、**match ipv6 address prefix-list**、および **match ip route-source prefix-list** コマンドでのみサポートされます。別の match 文は無視されます。

- 廃棄ルートには、アドミニストレーティブ ディスタンス 220 が常に割り当てられます。テーブル マップの設定は OSPF の廃棄ルートには適用されません。
- OSPF ルートのアドミニストレーティブ ディスタンスを設定する場合、**match route-type**、**match ipv6 address**、および **match ip route-source** コマンドの間に優先順位はありません。このように、Cisco NX-OS OSPF アドミニストレーティブ ディスタンスを設定するためのテーブル マップの動作は、Cisco IOS OSPF の場合と異なります。

- vPC コンフィギュレーション モードで **delay restore** *seconds* コマンドを設定する場合や、 マルチシャーシ EtherChannel トランク（MCT）上の VLAN がスイッチ仮想インターフェイス（SVI）を使用して OSPFv2 または OSPFv3 によって通知される場合、これらの SVI は設定された時間の間、vPC セカンダリ ノード上で MAX_LINK_COST で通知されます。その結果、すべてのルートまたはホストのプログラミングは、トラフィックを引き込む前に（セカンダリ vPC ノードのピア リロードで）vPC の同期操作後に完了します。この動作により、ノース サウス トラフィックのパケット損失を最小にできます。
- プライマリエリアとマルチエリアに同じエリア *ID* を設定すると、エラーが表示されずに設定が受け入れられます。プライマリエリアとマルチエリアを設定する場合は、同じエリア *ID* を使用しないでください。

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

- OSPF で **network** *ip address mask* コマンドを使用すると、エラー メッセージが表示され、**area** *area id* コマンドを使用してインターフェイスで OSPF を有効にするように求められます。
- OSPF のデフォルト タイマー（hello-interval:10 および dead-interval:40）を使用することをお勧めします。コンバージェンス時間を短縮するには、OSPF とともに BFD を使用できます。この組み合わせにより、1 秒未満のリンク/隣接フラップ検出と非常に短いコンバージェンス時間が実現します。
- OSPF サポートはアグレッシブタイマーですが、アグレッシブ イマーは調整をすぐにダウンさせ、CPU チャーンを引き起こすため、推奨されません。デフォルトのタイマーを使用し、BFD（双方向転送検出）を使用して 1 秒未満の障害検出を行うことを推奨します。
- Cisco NX-OS リリース 10.3(1)F 以降、OSPFv3 は Cisco Nexus 9808 スイッチでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、 OSPFv3 は Cisco Nexus 9804 スイッチでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、Cisco NX-OS スイッチの OSPFv3 暗号化および認証コマンドに対してキーチェーンのサポートが提供されます。
- Cisco NX-OS リリース 10.4(1)F 以降、OSPFv3 は Cisco Nexus 9808 および 9804 スイッチを搭載した Cisco Nexus X98900CD-A および X9836DM-A ライン カードでサポートされます。

デフォルト設定

次の表に、OSPFv3 パラメータのデフォルト設定値を示します。

表 *19 : OSPFv3* のデフォルト パラメータ

パラメータ	デフォルト
アドミニストレーティブ ディスタンス	110
hello 間隔	10 秒
デッド間隔	40 秒
廃棄ルート	イネーブル
グレースフル リスタートの猶予期間	60 秒
グレースフル リスタートの通知期間	15 秒
OSPFv3 機能	ディセーブル
スタブ ルータ アドバタイズメントの宣言期間	600 秒
リンク コスト計算の参照帯域幅	40 Gbps
LSA 最小到着時間	1000 ミリ秒
LSA グループ ペーシング	10 秒
SPF 計算初期遅延時間	200 ミリ秒
SPF 計算最小ホールド タイム	1000 ミリ秒
SPF 計算の最大待機時間	5000 ミリ秒

基本的なOSPFv3の設定

OSPFv3 は、OSPFv3 ネットワークを設計したあとに設定します。

OSPFv3の有効化

手順の概要

1. **configure terminal**
2. [**no**] **feature ospfv3**

3. （任意） **show feature**
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**[no] feature ospfv3** 例： `switch(config)# feature ospfv3`	OSPFv3 を有効にします。 このコマンドを持つ **no** キーワードを使用すると、OSPFv3 機能を無効にして、関連するすべての設定を削除します。
Step 3	（任意） **show feature** 例： `switch(config)# show feature`	有効および無効にされた機能を表示します。
Step 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

OSPFv3インスタンスの作成

OSPFv3 設定の最初のステップは、インスタンスまたはOSPFv3 インスタンスの作成です。作成した OSPFv3 インスタンスには、一意のインスタンス タグを割り当てます。インスタンス タグは任意の文字列です。各 OSPFv3 インスタンスには、省略可能な次のパラメータも設定できます。

- Router ID：この OSPFv3 インスタンスのルータ ID を設定します。このパラメータを使用しない場合は、ルータ ID 選択アルゴリズムが使用されます。「ルータ ID」セクションを参照してください。
- Administrative distance：ルーティング情報の送信元の信頼性をランク付けします。詳細については、「アドミニストレーティブ ディスタンス」のセクションを参照してください。
- Log adjacency changes：OSPFv3 ネイバーの状態が変化するたびにシステム メッセージを作成します。
- 名前のルックアップ：ローカル ホストのデータベースを検索または IPv6 の DNS 名を照会することでホスト名に OSPF ルータ ID を変換します。

- Maximum paths：OSPFv3 が、特定の宛先についてルート テーブルにインストールする同等パスの最大数を設定します。このパラメータは、複数パス間のロード バランシングに使用します。
- Reference bandwidth：ネットワークの算出 OSPFv3 コスト メトリックを制御します。算出コストは、参照帯域幅をインターフェイス帯域幅で割った値です。算出コストは、ネットワークが OSPFv3 インスタンスに追加されるときにリンク コストを割り当てると、無効にすることができます。詳細については、「OSPFv3 でのネットワークの設定」のセクションを参照してください。

OSPFv3 インスタンス パラメータの詳細については、「高度な OSPFv3 の設定」のセクションを参照してください。

始める前に

OSPFv3 機能が有効にされている必要があります（「OSPFv2 の有効化」のセクションを参照）。

使用する予定の OSPFv3 インスタンス タグが、このルータ上では使用されていないことを確認します。

show ospfv3 *instance-tag* を使用します。 コマンドを使用して、インスタンス タグが使用されていないことを確認します。

OSPFv3 がルータ ID（設定済みのループバック アドレスなど）を入手可能であるか、またはルータ ID オプションを設定する必要があります。

手順の概要

1. **configure terminal**
2. [**no**] **router ospfv3** *instance-tag*
3. （任意） **router-id** *ip-address*
4. （任意） **show ipv6 ospfv3** *instance-tag*
5. （任意） **log-adjacency-changes** [**detail**]
6. （任意） **passive-interface default**
7. （任意） **distance** *number*
8. （任意） **maximum-paths** *paths*
9. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的

Step 2	[**no**] **router ospfv3** *instance-tag*
	例：
	`switch(config)# router ospfv3 201`
	`switch(config-router)#`
		新規 OSPFv3 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。
		（注）
		インターフェイス モードでは、**no router ospfv3** *instance tag* コマンドによって OSPF の設定を削除できません。インターフェイス モードで設定された OSPFv3 コマンドはいずれも、手動で削除する必要があります。

Step 3	（任意） **router-id** *ip-address* 例： `switch(config-router)# router-id 192.0.2.1`	OSPFv3 ルータ ID を設定します。このドット付き 10 進表記の ID で、この OSPFv3 インスタンスが識別されます。この ID は、システムの設定済みインターフェイス上に存在する必要があります。
Step 4	（任意） **show ipv6 ospfv3** *instance-tag* 例： `switch(config-router)# show ipv6 ospfv3 201`	OSPFv3 情報を表示します。
Step 5	（任意） **log-adjacency-changes** [**detail**] 例： `switch(config-router)# log-adjacency-changes`	ネイバーの状態が変化するたびに、システムメッセージを生成します。
Step 6	（任意） **passive-interface default** 例： `switch(config-router)# passive-interface default`	すべてのインターフェイス上でルーティングが更新されないようにします。このコマンドは、VRF またはインターフェイス コマンド モードの設定によって上書きされます。
Step 7	（任意） **distance** *number* 例： `switch(config-router-af)# distance 25`	この OSPFv3 インスタンスのアドミニストレーティブ ディスタンスを設定します。範囲は 1 ～ 255 です。デフォルトは 110 です。
Step 8	（任意） **maximum-paths** *paths* 例： `switch(config-router-af)# maximum-paths 4`	ルート テーブル内の宛先に対する同等 OSPFv3 パスの最大数を設定します。指定できる範囲は 1 ～ 16 です。デフォルト値は 8 です。このコマンドはロードバランシングに使用されます。
Step 9	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次の例は、OSPFv3 インスタンスを作成する方法を示しています。

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

OSPFv3でのネットワークの設定

ルータがこのネットワークへの接続に使用するインターフェイスを介して、OSPFv3 へのネットワークを関連付けることで、このネットワークを設定できます（「ネイバー」セクションを参照）。すべてのネットワークをデフォルト バックボーン エリア（エリア 0）に追加したり、任意の 10 進数または IP アドレスを使用して新規エリアを作成したりできます。

（注） すべてのエリアは、バックボーン エリアに直接、または仮想リンク経由で接続する必要があります。

（注） インターフェイスの有効な IPv6 アドレスを設定するまでは、インターフェイス上で OSPFv3 がイネーブルになりません。

始める前に

OSPFv3 機能が有効にされている必要があります（「OSPFv2 の有効化」のセクションを参照）。

手順の概要

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. （任意） **show ipv6 ospfv3** *instance-tag* **interface** *interface-type slot/port*
6. （任意） **ospfv3 cost** *number*
7. （任意） **ospfv3 dead-interval** *seconds*
8. （任意） **ospfv3 hello-interval** *seconds*
9. （任意） **ospfv3 instance** *instance*
10. （任意） **ospfv3 mtu-ignore**
11. （任意） **ospfv3 network** {**broadcast** | **point-point**}
12. （任意） [**default** | **no**] **ospfv3 passive-interface**
13. （任意） **ospfv3 priority** *number*
14. （任意） **ospfv3 shutdown**

15. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**ipv6 address** *ipv6-prefix/length* 例： `switch(config-if)# ipv6 address 2001:0DB8::1/48`	このインターフェイスに IPv6 アドレスを割り当てます。
Step 4	**ipv6 router ospfv3** *instance-tag* **area** *area-id* [**secondaries none**] 例： `switch(config-if)# ipv6 router ospfv3 201 area 0`	OSPFv3 インスタンスおよびエリアにインターフェイスを追加します。
Step 5	（任意） **show ipv6 ospfv3** *instance-tag* **interface** *interface-type slot/port* 例： `switch(config-if)# show ipv6 ospfv3 201 interface ethernet 1/2`	OSPFv3 情報を表示します。
Step 6	（任意） **ospfv3 cost** *number* 例： `switch(config-if)# ospfv3 cost 25`	このインターフェイスの OSPFv3 コスト メトリックを設定します。デフォルトでは、参照帯域幅とインターフェイス帯域幅に基づいて、コスト メトリックが計算されます。有効な範囲は 1 ～ 65535 です。
Step 7	（任意） **ospfv3 dead-interval** *seconds* 例： `switch(config-if)# ospfv3 dead-interval 50`	OSPFv3 デッド間隔を秒単位で設定します。有効な範囲は 1 ～ 65535 です。デフォルトでは、hello 間隔の秒数の 4 倍です。
Step 8	（任意） **ospfv3 hello-interval** *seconds* 例：	OSPFv3 hello 間隔を秒単位で設定します。有効な範囲は 1 ～ 65535 です。デフォルトは 10 秒です。

	コマンドまたはアクション	目的
	`switch(config-if)# ospfv3 hello-interval 25`	
Step 9	（任意） **ospfv3 instance** *instance* 例： `switch(config-if)# ospfv3 instance 25`	OSPFv3 インスタンス ID を設定します。有効な範囲は 0 ～ 255 です。デフォルトは 0 です。インスタンス ID のスコープはリンクローカルです。
Step 10	（任意） **ospfv3 mtu-ignore** 例： `switch(config-if)# ospfv3 mtu-ignore`	OSPFv3 で、ネイバーとのあらゆる IP 最大伝送単位（MTU）不一致が無視されるよう設定します。デフォルトでは、ネイバー MTU がローカル インターフェイス MTU が不一致の場合には、隣接関係が確立されません。
Step 11	（任意） **ospfv3 network** {**broadcast** | **point-point**} 例： `switch(config-if)# ospfv3 network broadcast`	OSPFv3 ネットワーク タイプを設定します。
Step 12	（任意） [**default** | **no**] **ospfv3 passive-interface** 例： `switch(config-if)# ospfv3 passive-interface`	インターフェイス上でルーティングが更新されないようにします。このコマンドによって、ルータまたは VRF コマンド モードの設定が上書きされます。**default** オプションは、このインターフェイス モード コマンドを削除して、ルータまたは VRF の設定に戻します（設定がある場合）。
Step 13	（任意） **ospfv3 priority** *number* 例： `switch(config-if)# ospfv3 priority 25`	エリアの DR の決定に使用される OSPFv3 優先度を設定します。有効な範囲は 0 ～ 255 です。デフォルトは 1 です。「指定ルータ」の項を参照してください。
Step 14	（任意） **ospfv3 shutdown** 例： `switch(config-if)# ospfv3 shutdown`	このインターフェイス上の OSPFv3 インスタンスをシャットダウンします。
Step 15	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、OSPFv3 インスタンス 201 にネットワーク エリア 0.0.0.10 を追加する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
```

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

高度なOSPFv3の設定

OSPFv3 は、OSPFv3 ネットワークを設計したあとに設定します。

境界ルータのフィルタ リストの設定

OSPFv3 ドメインを、関連性のある各ネットワークを含む一連のエリアに分離できます。すべてのエリアは、エリア境界ルータ（ABR）経由でバックボーン エリアに接続している必要があります。OSPFv3 ドメインは、自律システム境界ルータ（ASBR）を介して、外部ドメインにも接続可能です。「エリア」の項を参照してください。

ABR には、省略可能な次の設定パラメータがあります。

- Area range：エリア間のルート集約を設定します。詳細については、「ルート集約の設定」の項を参照してください。
- Filter list：ABR 上で、外部エリアから受信したエリア間プレフィックス（タイプ 3）LSA をフィルタリングします。

ASBR もフィルタ リストをサポートしています。

始める前に

フィルタ リストが、着信または発信エリア間プレフィックス（タイプ 3）LSA の IP プレフィックスのフィルタリングに使用するルートマップを作成します。Route Policy Manager の設定（587 ページ）を参照してください。

手順の概要

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. （任意） **show ipv6 ospfv3 policy statistics area** *id* **filter-list** {**in** | **out**}
6. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospfv3** *instance-tag*
	例：
	`switch(config)# router ospfv3 201`
	`switch(config-router)#`
		インスタンス タグを設定して、新しい OSPFv3 インスタンスを作成します。

Step 3	**address-family ipv6 unicast**
	例：
	`switch(config-router)# address-family ipv6 unicast`
	`switch(config-router-af)#`
		IPv6 ユニキャスト アドレス ファミリ モードを開始します。

Step 4	**area** *area-id* **filter-list route-map** *map-name* {**in** | **out**} 例： `switch(config-router-af)# area 0.0.0.10 filter-list route-map FilterLSAs in`	ABR 上で着信または発信エリア間プレフィックス（タイプ 3）LSA をフィルタリングします。
Step 5	（任意） **show ipv6 ospfv3 policy statistics area** *id* **filter-list** {**in** | **out**} 例： `switch(config-router-af)# show ipv6 ospfv3 policy statistics area 0.0.0.10 filter-list in`	OSPFv3 ポリシー情報を表示します。
Step 6	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、ルート マップ用にフィルタを設定する例を示します。

```
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
```

スタブ エリアの設定

OSPFv3 ドメインの外部トラフィックが不要な個所にスタブ エリアを設定できます。スタブ エリアはAS外部（タイプ5）LSAをブロックし、不要な、選択したネットワークへの往復のルーティングを制限します。「スタブ エリア」の項を参照してください。また、すべての集約ルートがスタブ エリアを経由しないようブロックすることもできます。

始める前に

OSPF機能がイネーブルにされている必要があります（「OSPFv2のイネーブル化」の項を参照）。

設定されるスタブ エリア内に、仮想リンクと ASBR のいずれも含まれないことを確認します。

手順の概要

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

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospfv3** *instance-tag*
	例：
	`switch(config)# router ospfv3 201`
	`switch(config-router)#`
		新規 OSPFv3 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 3	**area** *area-id* **stub** 例： `switch(config-router)# area 0.0.0.10 stub`	このエリアをスタブ エリアとして作成します。
Step 4	（任意） **address-family ipv6 unicast** 例：	IPv6 ユニキャスト アドレス ファミリ モードを開始します。

	コマンドまたはアクション	目的
	`switch(config-router)# address-family ipv6 unicast` `switch(config-router-af)#`	
Step 5	（任意） **area** *area-id* **default cost** *cost* 例： `switch(config-router-af)# area 0.0.0.10 default-cost 25`	このスタブ エリアに送信されるデフォルト サマリ ルートのコスト メトリックを設定します。指定できる範囲は 0 ～ 16777215 です。
Step 6	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、すべてのサマリ ルート更新をブロックするスタブエリアを作成する例を示します。

```
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
```

Totally Stubby エリアの設定

Totally Stubby エリアを作成して、すべての集約ルート更新がスタブ エリアに入るのを防ぐことができます。

Totally Stubby エリアを作成するには、ルータ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **area** *area-id* **stub no-summary**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**area** *area-id* **stub no-summary** 例： `switch(config-router)# area 20 stub no-summary`	このエリアを Totally Stubby エリアとして作成します。

NSSA の設定

OSPFv3 ドメインの一部で一定限度の外部トラフィックが必要な場合は、その部分に NSSA を設定できます。また、この外部トラフィックを AS 外部（タイプ 5）LSA に変換して、このルーティング情報で OSPFv3 ドメインをフラッディングすることもできます。NSSA は、省略可能な次のパラメータで設定できます。

- No redistribution：NSSA をバイパスして OSPFv3 AS 内の他のエリアに到達するルートを再配布します。このオプションは、NSSA ASBR が ABR も兼ねているときに使用します。
- Default information originate：外部自律システムへのデフォルト ルートのタイプ 7 LSA を生成します。このオプションは、ASBR のルーティング テーブルにデフォルト ルートが含まれる場合に NSSA ASBR 上で使用します。このオプションは、ASBR のルーティング テーブルにデフォルト ルートが含まれるかどうかに関係なく、NSSA ASBR 上で使用できます。
- Route map：目的のルートのみが NSSA および他のエリア全体でフラッディングされるよう、外部ルートをフィルタリングします。
- No summary：すべての集約ルートが NSSA でフラッディングされないようにします。このオプションは NSSA ABR 上で使用します。
- Translate：NSSA 外のエリア向けに、タイプ 7 LSA を AS 外部 LSA（タイプ 5）に変換します。再配布されたルートを OSPFv3 自律システム全体でフラッディングするには、このコマンドを NSSA ABR 上で使用します。また、これらの AS 外部 LSA の転送アドレスを無効にすることもできます。

（注） 変換オプションでは、NSSA を作成し、他のオプションを設定する **area** *area-id* **nssa** コマンドの後に、別の **area** *area-id* **nssa** コマンドが必要です。

始める前に

OSPF 機能が有効にされている必要があります（「OSPFv2 の有効化 」の項を参照）。

設定する NSSA 上に仮想リンクがないことと、この NSSA がバックボーン エリアでないことを確認します。

手順の概要

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

7. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospfv3** *instance-tag*
	例：
	`switch(config)# router ospfv3 201`
	`switch(config-router)#`
		新規 OSPFv3 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 3	**area** *area-id* **nssa** [**no-redistribution**] [**default-information-originate**] [**route-map** *map-name*] [**no-summary**] 例： `switch(config-router)# area 0.0.0.10 nssa`	このエリアを NSSA として作成します。
Step 4	（任意） **area** *area-id* **nssa translate type7** {**always** | **never**} [**suppress-fa**] 例： `switch(config-router)# area 0.0.0.10 nssa translate type7 always`	AS 外部（タイプ 7）LSA を NSSA 外部（タイプ 5）LSA に変換するように NSSA を設定します。

Step 5	（任意） **address-family ipv6 unicast**
	例：
	`switch(config-router)# address-family ipv6 unicast`
	`switch(config-router-af)#`
		IPv6 ユニキャスト アドレス ファミリ モードを開始します。

Step 6	（任意） **area** *area-id* **default cost** *cost* 例： `switch(config-router-af)# area 0.0.0.10 default-cost 25`	この NSSA に送信されるデフォルト集約ルートのコスト メトリックを設定します。指定できる範囲は 0 ～ 16777215 です。
Step 7	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、すべての集約ルート更新をブロックする NSSA を作成する例を示します。

```
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
```

次に、デフォルト ルートを生成する NSSA を作成する例を示します。

```
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
```

次に、外部ルートをフィルタリングし、すべての集約ルート更新をブロックする NSSA を作成する例を示します。

```
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
```

次に、常に NSSA 外部（タイプ 5）LSA を AS 外部（タイプ 7）LSA に変換する NSSA を作成し NSSA を設定する例を示します。

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# area 0.0.0.10 nssa
switch(config-router)# area 0.0.0.10 nssa translate type 7 always
switch(config-router)# copy running-config startup-config
```

次に、すべての集約ルート更新をブロックする NSSA を作成する例を示します。

```
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
```

マルチエリアの隣接関係の設定

既存の OSPFv3 インターフェイスには複数のエリアを追加できます。追加の論理インターフェイスはマルチエリア隣接関係をサポートしています。

始める前に

OSPF 機能がイネーブルにされる必要があります（「OSPFv3 のイネーブル化 」の項を参照）。

インターフェイスにプライマリ エリアが設定されていることを確認します（「OSPFv3 のネットワークの設定」の項を参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*

3. **ipv6 router ospfv3** *instance-tag* **multi-area** *area-id*
4. （任意） **show ipv6 ospfv3** *instance-tag* **interface** *interface-type slot/port*
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**ipv6 router ospfv3** *instance-tag* **multi-area** *area-id* 例： `switch(config-if)# ipv6 router ospfv3 201 multi-area 3`	別のエリアにインターフェイスを追加します。
Step 4	（任意） **show ipv6 ospfv3** *instance-tag* **interface** *interface-type slot/port* 例： `switch(config-if)# show ipv6 ospfv3 201 interface ethernet 1/2`	OSPFv3 情報を表示します。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、OSPFv3 インターフェイスに別のエリアを追加する例を示します。

```
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
```

仮想リンクの設定

仮想リンクは、隔離されたエリアを中継エリアを介してバックボーンエリアに接続します。[仮想リンク]セクションを展開します。仮想リンクには、省略可能な次のパラメータを設定できます。

- Dead interval：ローカル ルータがデッドであることを宣言し、隣接関係を解消する前に、ネイバーが hello パケットを待つ時間を設定します。
- Hello interval：連続する hello パケット間の時間間隔を設定します。
- Retransmit interval：連続する LSA 間の推定時間間隔を設定します。
- Transmit delay：LSA をネイバーに送信する推定時間を設定します。

（注） リンクがアクティブになる前に、関与する両方のルータで仮想リンクを設定する必要があります。

始める前に

OSPF を有効にする必要があります（「OSPFv3 の有効化」のセクションを参照）。

手順の概要

1. **configure terminal**
2. **router ospfv3** *instance-tag*
3. **area** *area-id* **virtual-link** *router-id*
4. （任意） **show ipv6 ospfv3 virtual-link** [**brief**]
5. （任意） **dead-interval** *seconds*
6. （任意） **hello-interval** *seconds*
7. （任意） **retransmit-interval** *seconds*
8. （任意） **transmit-delay** *seconds*
9. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospfv3** *instance-tag* 例：	新規 OSPFv3 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

	コマンドまたはアクション	目的
	`switch(config)# router ospfv3 201` `switch(config-router)#`	

Step 3	**area** *area-id* **virtual-link** *router-id*
	例：
	`switch(config-router)# area 0.0.0.10 virtual-link 2001:0DB8::1`
	`switch(config-router-vlink)#`
		リモート ルータへの仮想リンクの端を作成します。仮想リンクをリモート ルータ上に作成して、リンクを完成させる必要があります。

Step 4	（任意） **show ipv6 ospfv3 virtual-link** [**brief**] 例： `switch(config-router-vlink)# show ipv6 ospfv3 virtual-link`	OSPFv3 仮想リンク情報を表示します。
Step 5	（任意） **dead-interval** *seconds* 例： `switch(config-router-vlink)# dead-interval 50`	OSPFv3デッド間隔を秒単位で設定します。有効な範囲は 1 ～ 65535 です。デフォルトでは、hello 間隔の秒数の 4 倍です。
Step 6	（任意） **hello-interval** *seconds* 例： `switch(config-router-vlink)# hello-interval 25`	OSPFv3 hello 間隔を秒単位で設定します。有効な範囲は 1 ～ 65535 です。デフォルトは 10 秒です。
Step 7	（任意） **retransmit-interval** *seconds* 例： `switch(config-router-vlink)# retransmit-interval 50`	OSPFv3 再送信間隔を秒単位で設定します。有効な範囲は 1 ～ 65535 です。デフォルトは 5 分です。
Step 8	（任意） **transmit-delay** *seconds* 例： `switch(config-router-vlink)# transmit-delay 2`	OSPFv3 送信遅延を秒単位で設定します。指定できる範囲は 1 ～ 450 です。デフォルトは 1 です。
Step 9	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、2 つの ABR 間に簡単な仮想リンクを作成する例を示します。

ABR 1（ルータ ID 2001:0DB8::1）の設定は、次のとおりです。

```
switch# configure terminal
switch(config)# router ospfv3 201
```

```
switch(config-router)# area 0.0.0.10 virtual-link 2001:0DB8::10
switch(config-router-vlink)# copy running-config startup-config
```

ABR 2（ルータ ID 2001:0DB8::10）の設定は、次のとおりです。

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# area 0.0.0.10 virtual-link 2001:0DB8::1
switch(config-router-vlink)# copy running-config startup-config
```

再配布の設定

他のルーティング プロトコルから学習したルートを、ASBR 経由で OSPFv3 自律システムに再配布できます。

OSPF でのルート再配布には、省略可能な次のパラメータを設定できます。

- Default information originate：外部自律システムへのデフォルト ルートの AS 外部（タイプ 5）LSA を生成します。

（注） Default information originate はオプションのルート マップ内の **match** 文を無視します。

- Default metric：すべての再配布ルートに同じコスト メトリックを設定します。

（注） スタティック ルートを再配布する場合、デフォルトの 7.0(3)I7(6) スタティック ルートを正常に再配布するためには、Cisco NX-OS は **default-information originate** コマンドを必要とします。

始める前に

OSPF 機能が有効にされている必要があります（「OSPFv2 の有効化 」の項を参照）。

再配布で使用する、必要なルート マップを作成します。

手順の概要

1. **configure terminal**
2. **router ospfv3** *instance-tag*
3. **address-family ipv6 unicast**
4. **redistribute** {**bgp***id* | **direct** | **isis** *id* | **rip** *id* | **static** | **dhcpv6**} **route-map** *map-name*
5. **default-information originate** [**always**] [**route-map** *map-name*]
6. **default-metric** *cost*
7. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospfv3** *instance-tag*
	例：
	`switch(config)# router ospfv3 201`
	`switch(config-router)#`
		新規 OSPFv3 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 3	**address-family ipv6 unicast**
	例：
	`switch(config-router)# address-family ipv6 unicast`
	`switch(config-router-af)#`
		IPv6 ユニキャスト アドレス ファミリ モードを開始します。

Step 4	**redistribute** {**bgp***id* | **direct** | **isis** *id* | **rip** *id* | **static** | **dhcpv6**} **route-map** *map-name* 例： `switch(config-router-af)# redistribute bgp route-map FilterExternalBGP`	設定したルート マップ経由で、選択したプロトコルを OSPFv3 に再配布します。 （注） スタティック ルートを再配布する場合、デフォルトの 7.0(3)I7(6) スタティック ルートを正常に再配布するためには、Cisco NX-OS は **default-information originate** コマンドを必要とします。
Step 5	**default-information originate** [**always**] [**route-map** *map-name*] 例： `switch(config-router-af)# default-information-originate route-map DefaultRouteFilter`	デフォルトのルートが RIB に存在する場合、この OSPFv3 ドメインにデフォルトのルートを作成します。次の省略可能なキーワードを使用します。 • **always**：ルートが RIB に存在しない場合でも、常にデフォルト ルート 0.0.0. を生成します。 • **route-map**：ルート マップが true を返す場合にデフォルト ルートを生成します。 （注） このコマンドは、ルートマップの **match** 文を無視します。
Step 6	**default-metric** *cost* 例：	再配布されたルートのコスト メトリックを設定します。指定できる範囲は 1 ～ 16777214 です。このコマンドは、直接接続されたルートには適用されません。

	コマンドまたはアクション	目的
	`switch(config-router-af)# default-metric 25`	ルート マップを使用して、直接接続されたルートのデフォルトのメトリックを設定します。
Step 7	（任意） **copy running-config startup-config** 例： `switch(config-router-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、ボーダー ゲートウェイ プロトコル（BGP）をOSPFv3に再配布する例を示します。

```
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
```

再配布されるルート数の制限

ルート再配布によって、OSPFv3 ルート テーブルに多数のルートを追加できます。外部プロトコルから受け取るルートの数の上限を設定できます。OSPFv3には、再配布されるルート制限を設定するための次のオプションがあります。

- 上限固定：設定された最大値にOSPFv3が達すると、メッセージをログに記録します。OSPFv3はそれ以上の再配布されたルートを受け付けません。任意で、最大値のしきい値パーセンテージを設定して、OSPFv3がこのしきい値を超えたときに警告を記録するようにすることもできます。
- 警告のみ：OSPFv3 が最大値に達したときのみ、警告のログを記録します。OSPFv3 は、再配布されたルートを受け入れ続けます。
- 取り消し：OSPFv3が最大値に達したときに設定したタイムアウト期間を開始します。このタイムアウト期間後、現在の再配布されたルート数が最大制限より少なければ、OSPFv3はすべての再配布されたルートを要求します。再配布されたルートの現在数が最大数に達した場合、OSPFv3 はすべての再配布されたルートを取り消します。OSPFv3 が追加の再配布されたルートを受け付ける前に、この状況を解消する必要があります。任意で、タイムアウト期間を設定できます。

始める前に

OSPF 機能が有効にされている必要があります（「OSPFv2 の有効化 」の項を参照）。

手順の概要

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 timemout*]]
6. （任意） **show running-config ospfv3**
7. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospfv3** *instance-tag*
	例：
	`switch(config)# router ospfv3 201`
	`switch(config-router)#`
		新規 OSPFv3 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 3	**address-family ipv6 unicast**
	例：
	`switch(config-router)# address-family ipv6 unicast`
	`switch(config-router-af)#`
		IPv6 ユニキャスト アドレス ファミリ モードを開始します。

Step 4	**redistribute** {**bgp***id* | **direct** | **isis** *id* | **rip** *id* | **static**} **route-map** *map-name* 例： `switch(config-router-af)# redistribute bgp route-map FilterExternalBGP`	設定したルート マップ経由で、選択したプロトコルを OSPFv3 に再配布します。
Step 5	**redistribute maximum-prefix***max* [*threshold*] [**warning-only** | **withdraw** [*num-retries timemout*]] 例： `switch(config-router-af)# redistribute maximum-prefix 1000 75 warning-only`	OSPFv2 が配布するプレフィックスの最大数を指定します。指定できる範囲は 0 ～ 65536 です。任意で次のオプションを指定します。 • *threshold*：警告メッセージをトリガーする最大プレフィックスの割合。 • **warning-only**：プレフィックスの最大数を超えた場合に警告メッセージを記録します。 • **withdraw**：再配布されたすべてのルートを取り消し、任意で再配布されたルートを取得しようと試みます。*num-retries* の範囲は 1 ～ 12 です。

	コマンドまたはアクション	目的
		timeout の範囲は 60 ～ 600 秒です。デフォルトは 300 秒です。
Step 6	（任意） **show running-config ospfv3** 例： `switch(config-router-af)# show running-config ospf`	OSPFv3 設定を表示します。
Step 7	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、OSPF に再配布されるルートの数を制限する例を示します。

```
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
```

ルート集約の設定

サマリ アドレス範囲を設定することで、エリア間ネットワークのルート集約を設定できます。また、ASBR 上のこれらのルートのサマリ アドレスを設定して、外部の再配布されたルートのルート集約を設定することもできます。詳細については、「ルート集約」を参照してください。

始める前に

OSPF 機能が有効にされている必要があります（「OSPFv2 の有効化 」の項を参照）。

手順の概要

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*]
5. **summary-address** *ipv6-prefix/length* [**no-advertise**] [**tag** *tag*]
6. （任意） **show ipv6 ospfv3 summary-address**
7. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospfv3** *instance-tag*
	例：
	`switch(config)# router ospfv3 201`
	`switch(config-router)#`
		新規 OSPFv3 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 3	**address-family ipv6 unicast**
	例：
	`switch(config-router)# address-family ipv6 unicast`
	`switch(config-router-af)#`
		IPv6 ユニキャスト アドレス ファミリ モードを開始します。

Step 4	**area** *area-id* **range** *ipv6-prefix/length* [**no-advertise**] [**cost** *cost*] 例： `switch(config-router-af)# area 0.0.0.10 range 2001:0DB8::/48 advertise`	一定の範囲のアドレスのサマリ アドレスを ABR 上に作成します。このサマリ アドレスをエリア間プレフィックス（タイプ 3）LSA にアドバタイズすることもできます。cost の範囲は 0 ～ 16777215 です。
Step 5	**summary-address** *ipv6-prefix/length* [**no-advertise**] [**tag** *tag*] 例： `switch(config-router-af)# summary-address 2001:0DB8::/48 tag 2`	一定の範囲のアドレスのサマリ アドレスを ABR 上に作成します。ルート マップによる再配布で使用できるよう、このサマリ アドレスにタグを割り当てることもできます。
Step 6	（任意） **show ipv6 ospfv3 summary-address** 例： `switch(config-router-af)# show ipv6 ospfv3 summary-address`	OSPFv3 サマリ アドレスに関する情報を表示します。
Step 7	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、ABR 上のエリア間のサマリ アドレスを作成する例を示します。

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# address-family ipv6 unicast
switch(config-router-af)# area 0.0.0.10 range 2001:0DB8::/48
switch(config-router-af)# copy running-config startup-config
```

次に、ASBR 上のサマリ アドレスを作成する例を示します。

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# address-family ipv6 unicast
switch(config-router-af)# summary-address 2001:0DB8::/48
switch(config-router-af)# no discard route internal
switch(config-router-af)# copy running-config startup-config
```

ルートのアドミニストレーティブ ディスタンスの設定

OSPFv3 によって RIB に追加されるルートのアドミニストレーティブ ディスタンスを設定できます。

アドミニストレーティブ ディスタンスは、ルーティング情報源の信頼性を示す評価基準です。値が高いほど信頼性の評価は低くなります。一般的にルートは、複数のルーティング プロトコルを通じて検出されます。アドミニストレーティブディスタンスは、複数のルーティングプロトコルから学習したルートを区別するために使用されます。最もアドミニストレーティブ ディスタンスが低いルートが IP ルーティング テーブルに組み込まれます。

始める前に

OSPF が有効になっていることを確認します（OSPFv3 の設定（169 ページ）セクションを参照）。

「OSPFv3 の注意事項および制約事項（184 ページ）」のセクションにあるこの機能の注意事項と制限事項を参照してください。

手順の概要

1. **configure terminal**
2. **router ospfv3** *instance-tag*
3. **address-family ipv6 unicast**
4. [**no**] **table-map** *map-name*
5. **exit**
6. **exit**
7. **route-map** *map-name* [**permit** | **deny**] [*seq*]
8. **match route-type** *route-type*
9. **match ip route-source prefix-list** *name*
10. **match ipv6 address prefix-list** *name*
11. **set distance** *value*
12. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospfv3** *instance-tag*
	例：
	`switch(config)# router ospfv3 201`
	`switch(config-router)#`
		新規 OSPFv3 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 3	**address-family ipv6 unicast**
	例：
	`switch(config-router)# address-family ipv6 unicast`
	`switch(config-router-af)#`
		IPv6 ユニキャスト アドレス ファミリ モードを開始します。

Step 4	[**no**] **table-map** *map-name* 例： `switch(config-router-af)# table-map foo`	OSPFv3 ルートを RIB に送信する前に、OSPFv2 ルートをフィルタリングまたは変更するポリシーを設定します。マップ名には最大 63 文字の英数字を入力できます。

Step 5	**exit**
	例：
	`switch(config-router-af)# exit`
	`switch(config-router)#`
		ルータ アドレス ファミリ コンフィギュレーション モードを終了します。

Step 6	**exit**
	例：
	`switch(config-router)# exit`
	`switch(config)#`
		ルータ コンフィギュレーション モードを終了します。

Step 7	**route-map** *map-name* [**permit** | **deny**] [*seq*]
	例：
	`switch(config)# route-map foo permit 10`
	`switch(config-route-map)#`
		ルート マップを作成するか、または既存のルート マップに対応するルート マップ設定モードを開始します。ルート マップのエントリを順序付けるには、*seq* を使用します。
		（注）
		permit オプションで、ディスタンスを設定することができます。**deny** オプションを使用すると、デフォルトのディスタンスが適用されます。

	コマンドまたはアクション	目的
Step 8	**match route-type** *route-type* 例： `switch(config-route-map)# match route-type external`	次のルート タイプのいずれかと照合します。 • external：外部ルート（BGP、EIGRP、OSPF タイプ 1 または 2） • エリア間：OSPF エリア間ルート • internal：内部ルート（OSPF エリア内またはエリア間ルートを含む） • エリア内：OSPF のエリア内ルート • nssa-external：NSSA 外部ルート（OSPF タイプ 1 または 2） • type-1：OSPF 外部タイプ 1 ルート • type-2：OSPF 外部タイプ 2 ルート
Step 9	**match ip route-source prefix-list** *name* 例： `switch(config-route-map)# match ip route-source prefix-list p1`	1 つまたは複数の IP プレフィックス リストに対して、ルートの IPv6 ルート送信元アドレスまたはルータ ID と照合します。プレフィックス リストは **ip prefix-list** コマンドを使用して作成します。 （注） OSPFv3 では、ルータ ID は 4 バイトです。
Step 10	**match ipv6 address prefix-list** *name* 例： `switch(config-route-map)# match ipv6 address prefix-list p1`	1 つまたは複数の IPv6 プレフィックス リストと照合。プレフィックス リストは **ip prefix-list** コマンドを使用して作成します。
Step 11	**set distance** *value* 例： `switch(config-route-map)# set distance 150`	OSPFv3 のルートのアドミニストレーティブ ディスタンスを設定します。範囲は 1 ～ 255 です。
Step 12	（任意） **copy running-config startup-config** 例： `switch(config-route-map)# copy running-config startup-config`	この設定変更を保存します。

例

次に、OSPFv3 アドミニストレーティブ ディスタンスについて、エリア間ルートを 150、外部ルートを 200、およびプレフィックス リスト p1 内のすべてのプレフィックスを 190 に設定する例を示します。

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# address-family ipv6 unicast
switch(config-router-af)# table-map foo
switch(config-router)# exit
switch(config)# exit
switch(config)# route-map foo permit 10
switch(config-route-map)# match route-type inter-area
switch(config-route-map)# set distance 150
switch(config)# route-map foo permit 20
switch(config-route-map)# match route-type external
switch(config-route-map)# set distance 200
switch(config)# route-map foo permit 30
switch(config-route-map)# match ip route-source prefix-list p1
switch(config-route-map)# match ipv6 address prefix-list p1
switch(config-route-map)# set distance 190
switch(config-route-map)# copy running-config startup-config
```

デフォルト タイマーの変更

OSPFv3 には、プロトコル メッセージの動作および最短パス優先（SPF）の計算を制御する多数のタイマーが含まれています。OSPFv3 には、省略可能な次のタイマー パラメータが含まれます。

- LSA arrival time：ネイバーから着信する LSA 間で許容される最小間隔を設定します。この時間より短時間で到着する LSA はドロップされます。
- Pacing LSAs：LSA が集められてグループ化され、リフレッシュされて、チェックサムが計算される間隔、つまり期限切れとなる間隔を設定します。このタイマーは、LSA 更新が実行される頻度を制御し、LSA 更新メッセージで送信される LSA 更新の数を制御します（「フラッディングと LSA グループ ペーシング」を参照）。
- Throttle LSAs：LSA 生成のレート制限を設定します。このタイマーは、トポロジが変更された後に LSA が生成される頻度を制御します。
- Throttle SPF calculation：SPF 計算の実行頻度を制御します。

インターフェイス レベルでは、次のタイマーも制御できます。

- Retransmit interval：連続する LSA 間の推定時間間隔を設定します。
- Transmit delay：LSA をネイバーに送信する推定時間を設定します。

hello 間隔とデッド タイマーに関する情報の詳細については、「OSPFv3 のネットワークの設定」の項を参照してください。

手順の概要

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. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospfv3** *instance-tag*
	例：
	`switch(config)# router ospfv3 201`
	`switch(config-router)#`
		新規 OSPFv3 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 3	**timers lsa-arrival** *msec* 例： `switch(config-router)# timers lsa-arrival 2000`	LSA 到着時間をミリ秒で設定します。範囲は 10 ～ 600000 です。デフォルトは 1000 ミリ秒です。
Step 4	**timers lsa-group-pacing** *seconds* 例： `switch(config-router)# timers lsa-group-pacing 200`	LSA がグループ化される間隔を秒で設定します。範囲は 1 ～ 1800 です。デフォルトは 10 秒です。
Step 5	**timers throttle lsa** *start-time hold-interval max-time* 例： `switch(config-router)# timers throttle lsa network 350 5000 6000`	LSA 生成のレート制限をミリ秒で設定します。次のタイマーを設定できます。 • *start-time*：指定できる範囲は 0 ～ 5000 ミリ秒です。デフォルト値は 0 ミリ秒です。 • *hold-interval*：指定できる範囲は 50 ～ 30,000 ミリ秒です。デフォルト値は 5000 ミリ秒です。 • *max-time*：指定できる範囲は 50 ～ 30,000 ミリ秒です。デフォルト値は 5000 ミリ秒です。
Step 6	**address-family ipv6 unicast** 例：	IPv6 ユニキャスト アドレス ファミリ モードを開始します。

	コマンドまたはアクション	目的
	`switch(config-router)# address-family ipv6 unicast` `switch(config-router-af)#`	
Step 7	**timers throttle spf** *delay-time hold-time max-time* 例： `switch(config-router-af)# timers throttle spf 3000 2000`	SPF 最適パス スケジュールを次のタイマを使用して、SPF 最適パス計算間（秒単位）で設定します。 • *delay-time*：範囲は 1 ～ 600000 ミリ秒です。デフォルトは 200 ミリ秒です。 • *hold-time*：範囲は 1 ～ 600000 ミリ秒です。デフォルト値は、1000 ミリ秒です。 • *max-wait*：範囲は 1 ～ 600000 ミリ秒です。デフォルト値は 5000 ミリ秒です。

Step 8	**interface** *type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 9	**ospfv3 retransmit-interval** *seconds* 例： `switch(config-if)# ospfv3 retransmit-interval 30`	このインターフェイスから送信される各 LSA 間の推定時間間隔を設定します。有効な範囲は1～65535です。デフォルトは 5 分です。
Step 10	**ospfv3 transmit-delay** *seconds* 例： `switch(config-if)# ospfv3 transmit-delay 600`	LSA をネイバーに送信する推定時間間隔を秒で設定します。指定できる範囲は 1 ～ 450 です。デフォルトは 1 です。
Step 11	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップコンフィギュレーションにコピーします

例

次に、lsa-group-pacing オプションで LSA フラッディングを制御する例を示します。

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# timers lsa-group-pacing 300
switch(config-router)# copy running-config startup-config
```

グレースフル リスタートの設定

デフォルトでは、グレースフル リスタートは有効です。OSPFv3 インスタンスのグレースフル リスタートには、省略可能な次のパラメータを設定できます。

- Grace period：グレースフル リスタートの開始後に、ネイバーが隣接関係を解消するまでに待つ時間を設定します。
- Helper mode disabled：ローカル OSPFv3 インスタンスのヘルパー モードをディセーブルにします。OSPFv3 は、ネイバーのグレースフル リスタートには関与しません。
- Planned graceful restart only：予定された再起動の場合にのみグレースフル リスタートがサポートされるよう、OSPFv3 を設定します。

始める前に

OSPFv3 機能が有効にされている必要があります（「OSPFv2 の有効化 」の項を参照）。

すべてのネイバーで、一致した省略可能なパラメータ一式とともにグレースフル リスタートが設定されていることを確認します。

手順の概要

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. （任意） **show ipv6 ospfv3** *instance-tag*
8. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospfv3** *instance-tag*
	例：
	`switch(config)# router ospfv3 201`
	`switch(config-router)#`
		新規 OSPFv3 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

	コマンドまたはアクション	目的
Step 3	**graceful-restart** 例： `switch(config-router)# graceful-restart`	グレースフル リスタートを有効にします。グレースフル リスタートは、デフォルトで有効にされています。
Step 4	**graceful-restart grace-period** *seconds* 例： `switch(config-router)# graceful-restart grace-period 120`	猶予期間を秒で設定します。範囲は 5 ～ 1800 秒です。デフォルトは 60 秒です。
Step 5	**graceful-restart helper-disable** 例： `switch(config-router)# graceful-restart helper-disable`	ヘルパー モードを無効にします。デフォルトでは、イネーブルです。
Step 6	**graceful-restart planned-only** 例： `switch(config-router)# graceful-restart planned-only`	予定された再起動時にのみグレースフル リスタートを設定します。
Step 7	（任意） **show ipv6 ospfv3** *instance-tag* 例： `switch(config-router)# show ipv6 ospfv3 201`	OPSFv3 情報を表示します。
Step 8	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、ディセーブルにされているグレースフル リスタートをイネーブルにし、猶予期間を 120 秒に設定する例を示します。

```
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
```

OSPFv3 インスタンスの再起動

OSPv3インスタンスを再起動できます。この処理では、インスタンスのすべてのネイバーが消去されます。

OSPFv3 インスタンスを再起動して、関連付けられたすべてのネイバーを削除するには、次のコマンドを使用します。

手順の概要

1. **restart ospfv3** *instance-tag*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**restart ospfv3** *instance-tag* 例： `switch(config)# restart ospfv3 201`	OSPFv3 インスタンスを再起動して、すべてのネイバーを削除します。

仮想化による OSPFv3 の設定

複数 OSPFv3 インスタンスを設定できます。各仮想デバイス コンテキスト（VDC）内に複数の VRF を作成して、各 VRF で同じまたは複数の OSPFv3 インスタンスを使用することもできます。VRF には OSPFv3 インターフェイスを割り当てます。

（注） インターフェイスの VRF を設定した後に、インターフェイスの他のすべてのパラメータを設定します。インターフェイスの VRF を設定すると、そのインターフェイスのすべての設定が削除されます。

始める前に

OSPFv3 機能が有効にされている必要があります（「OSPFv2 の有効化 」の項を参照）。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **router ospfv3** *instance-tag*
4. **vrf** *vrf-name*
5. （任意） **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. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context RemoteOfficeVRF`
	`switch(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。

Step 3	**router ospfv3** *instance-tag*
	例：
	`switch(config)# router ospfv3 201`
	`switch(config-router)#`
		新規 OSPFv3 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 4	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf RemoteOfficeVRF`
	`switch(config-router-vrf)#`
		ルータ VRF 設定モードを開始します。

Step 5	（任意） **maximum-paths** *paths* 例： `switch(config-router-vrf)# maximum-paths 4`	この VRF のルート テーブル内の宛先への、同じ OSPFv3 パスの最大数を設定します。このコマンドはロード バランシングに使用します。

Step 6	**interface** *type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 7	**vrf member** *vrf-name* 例： `switch(config-if)# vrf member RemoteOfficeVRF`	このインターフェイスを VRF に追加します。
Step 8	**ipv6 address** *ipv6-prefix/length* 例： `switch(config-if)# ipv6 address 2001:0DB8::1/48`	このインターフェイスの IP アドレスを設定します。このステップは、このインターフェイスを VRF に割り当てたあとに行う必要があります。

	コマンドまたはアクション	目的
Step 9	**ipv6 ospfv3** *instance-tag* **area** *area-id* 例： `switch(config-if)# ipv6 ospfv3 201 area 0`	設定した OSPFv3 インスタンスおよびエリアに、このインターフェイスを割り当てます。
Step 10	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップコンフィギュレーションにコピーします

例

次に、VRF を作成して、その VRF にインターフェイスを追加する例を示します。

```
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
```

暗号化および認証の構成

Cisco Nexus リリース 10.2（1）以降では、ESP カプセル化を使用して OSPFv3 メッセージを暗号化および認証できます。OSPFv3 は、セキュア接続を IPSec に依存しています。IPSec は、次の 2 つのカプセル化タイプをサポートしています。

- 認証ヘッダー（AH）
- Encapsulating Security Payload（ESP）
- RFC4552「Authentication/Confidentiality for OSPFv3」は、上記の両方の側面をカバーしています。

ESP設定は、OSPFv3 メッセージの暗号化と認証の両方を提供します。

Cisco Nexus リリース 10.4(1)F 以降では、キーチェーン オプションを使用して暗号化および認証アルゴリズムとキーを構成できます。

制限事項は次のとおりです。

1. IPSec トランスポートモードのみがサポートされ、トンネルモードはサポートされません。

2. AH と ESP の設定は、インターフェイス上では一緒に使用できません。ただし、2 つの異なるインターフェイスに AH と ESP を設定できます。
3. RFC 4552 のセクション 10 で定義されている中断のないキー再生成はサポートされていません。
4. 次の暗号化アルゴリズムが ESP でサポートされます。
   - AES-CBC（128 ビット）
   - AES 192 ビットと AES 256 ビットは、このリリースではサポートされません。
   - 3DES-CBC
   - NULL
5. ESP では次の認証がサポートされます。
   - SHA-1
   - NULL
6. 1 つの ESP CLI で暗号化アルゴリズムと認証アルゴリズムの両方を NULL に設定することはできません。
7. 複数のエリアの一部であるインターフェイスは、親と同じ ESP パラメータを使用します。
8. 設定中に SPI が競合すると、エラーがユーザにスローされ、設定は保存されません。そのため、ESP 構成を変更する場合は、ユーザーは新しい構成に異なる SPI を使用する必要があります。
9. 最大 128 の SA/SPI 値を OSPFv3 プロセスごとに設定できます。

次のレベルで ESP を設定できます。

- ルータ
- エリア
- インターフェイス
- 仮想リンク

ルータ レベルでの OSPFv3 暗号化の設定

次のコマンドを使用して、ルータ レベルでOSPFv3パケットを暗号化および認証するようにOSPFv3 ESP を設定できます。

キーチェーンの構成方法に関する詳細は、『**Cisco Nexus 9000 シリーズ NX-OS セキュリティ構成ガイド**』の「キーチェーン管理の構成」を参照してください。

始める前に

OSPFv3 機能を有効にします。

認証パッケージを有効にします。

手順

Step 1 グローバル設定モードを開始します。

switch# **configure terminal**

Step 2 OSPFv3を有効にします。

switch(config)# **feature ospfv3**

Step 3 認証パッケージを有効にします。

switch(config)# **feature imp**

Step 4 インスタンスタグが設定された新しい OSPFv3 インスタンスを作成します。

switch(config)# **router ospfv3** *instance-tag*

Step 5 IPSec ESP 暗号化を有効にします:

switch(config-router)# **encryption ipsec spi** *spi_id* **esp** [*encrypt_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *enc_keychain_name* | **null**] **authentication** [*auth_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *auth_keychain_name* | **null**]

spi_id を使用してセキュリティポリシーインデックスを指定し、*encrypt_algorithm* を使用して暗号化アルゴリズムを定義できます。3DES、AES 128、または null を指定できます。番号 0、3、および 7 は、*key* の形式を指定します。認証アルゴリズムは、*auth_algorithm*（SHA-1 または NULL）で定義できます。

key-chain オプションを使用して、キーとアルゴリズムも構成できます。

Step 6 （任意）OSPFv3 情報を表示します。

switch(config)# **show running-config ospfv3**

エリア レベルでの OSPFv3 暗号化の設定

次のコマンドを使用して、エリアレベルでOSPFv3パケットを暗号化および認証するようにOSPFv3 ESPを設定できます。

キーチェーンの構成方法に関する詳細は、『**Cisco Nexus 9000 シリーズ NX-OS セキュリティ構成ガイド**』の「キーチェーン管理の構成」を参照してください。

始める前に

OSPFv3 機能を有効にします。

認証パッケージを有効にします。

手順

Step 1 グローバル設定モードを開始します。

switch# **configure terminal**

Step 2 OSPFv3を有効にします。

switch(config)# **feature ospfv3**

Step 3 認証パッケージを有効にします。

switch(config)# **feature imp**

Step 4 インスタンスタグが設定された新しい OSPFv3 インスタンスを作成します。

switch(config)# **router ospfv3** *instance-tag*

Step 5 IPSec ESP 暗号化を有効にします:

switch(config-router)#**area** *area-num* **encryption ipsec spi** *spi_val* **esp** *encrypt_algorithm* [**0** | **3** | **7***key* | **key-chain** *enc_keychain_name* | **null**] **authentication** *auth_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *auth_keychain_name* | **null**]

spi_id を使用してセキュリティポリシーインデックスを指定し、*encrypt_algorithm* を使用して暗号化アルゴリズムを定義できます。3DES、AES 128、または null を指定できます。番号 0、3、6 および 7 は、*key* の形式を指定します。認証アルゴリズムは、*auth_algorithm*（SHA-1 または NULL またはキーチェーン）で定義できます。

key-chain オプションを使用して、キーとアルゴリズムも構成できます。

Step 6 （任意）OSPFv3 情報を表示します。

switch(config)# **show running-config ospfv3**

インターフェイスレベルでの OSPFv3 暗号化の設定

次のコマンドを使用して、インターフェイス レベルで OSPFv3 パケットを暗号化および認証するように OSPFv3 ESP を設定できます。

キーチェーンの構成方法に関する詳細は、『**Cisco Nexus 9000 シリーズ NX-OS セキュリティ構成ガイド**』の「キーチェーン管理の構成」を参照してください。

始める前に

OSPFv3 をイネーブルにする必要があります。

認証パッケージを有効にします。

手順

Step 1 グローバル設定モードを開始します。

switch# **configure terminal**

Step 2 OSPFv3を有効にします。

switch(config)# **feature ospfv3**

Step 3 認証モードをイネーブルにします。

switch(config)# **feature imp**

Step 4 イーサネット インターフェイス設定モードを開始します:

switch(config)# **interface ethernet** *interface*

Step 5 インターフェイスのOSPFv3インスタンスとエリアを指定します。

switch（config-if）**#** instance-tag area-id **ipv6 router ospfv3 area**

Step 6 IPSec ESP 暗号化を有効にします:

switch(config-if)# **ospfv3 encryption ipsec spi** *spi_id* **esp** *encrypt_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *enc_keychain_name* | **null**] **authentication** *auth_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *auth_keychain_name* | **null**]

spi_id を使用してセキュリティポリシーインデックスを指定し、*encrypt_algorithm* を使用して暗号化アルゴリズムを定義できます。3DES、AES 128、または null を指定できます。番号 0、3、および 7 は、*key* の形式を指定します。認証アルゴリズムは、*auth_algorithm* （SHA-1 または NULL）で定義できます。

key-chain オプションを使用して、キーとアルゴリズムを構成することもできます。

Step 7 （オプション）インターフェイスの実行設定を表示します:

switch(config-if)#**show run interface** *interface*

設定例

次に、イーサネット インターネット 3/2 の セキュリティ を有効にする例を示します。

```
switch# configure terminal
switch(config)# feature ospfv3
switch(config)# feature imp
switch(config)# interface ethernet 3/2
switch(config-if)# ipv6 router ospfv3 1 area 0.0.0.0
switch(config-if)# ospfv3 encryption ipsec spi 444
  esp  Specify encryption parameters
switch(config-if)# ospfv3 encryption ipsec spi 444 esp
  3des  Use the triple DES algorithim
  aes   Use the AES algorithim
  key-chain  Encryption password key-chain
  null  Use NULL authentication
switch(config-if)# ospfv3 encryption ipsec spi 444 esp aes
```

```
  128  Use the 128-bit AES algorithim
switch(config-if)# ospfv3 encryption ipsec spi 444 esp aes 128
  0     Specifies an UNENCRYPTED encryption key will follow
  3     Specifies an 3DES ENCRYPTED encryption key will follow
  7     Specifies a Cisco type 7  ENCRYPTED encryption key will follow
  WORD  The UNENCRYPTED (cleartext) encryption key
switch(config-if)# ospfv3 encryption ipsec spi 444 esp aes 128
12345678123456781234567812345678 authentication null
switch(config-if)# sh ospfv3 interface
 Ethernet3/2 is up, line protocol is up
    IPv6 address 1:1:1:1::2/64
    Process ID 1 VRF default, Instance ID 0, area 0.0.0.0
    Enabled by interface configuration
    State DOWN, Network type BROADCAST, cost 40
    ESP Encryption AES, Authentication NULL, SPI 444, ConnId 444
switch(config-if)#
```

仮想リンクの OSPFv3 暗号化の設定

次のコマンドを使用して、仮想リンクのOSPFv3パケットを暗号化および認証するようにOSPFv3 ESP を設定できます。

キーチェーンの構成方法に関する詳細は、『**Cisco Nexus 9000 シリーズ NX-OS セキュリティ構成ガイド**』の「キーチェーン管理の構成」を参照してください。

始める前に

OSPFv3 機能を有効にします。

認証パッケージを有効にします。

手順

Step 1 グローバル設定モードを開始します。

switch# **configure terminal**

Step 2 OSPFv3を有効にします。

switch(config)# **feature ospfv3**

Step 3 認証パッケージを有効にします。

switch(config)# **feature imp**

Step 4 インスタンスタグが設定された新しい OSPFv3 インスタンスを作成します。

switch(config)#**router ospfv3** *instance-tag*

Step 5 リモート ルータへの仮想リンクの端を作成します。仮想リンクをリモート ルータ上に作成して、リンクを完成させる必要があります。

switch(config-router)# **area** *area-id* **virtual-link** *router-id*

Step 6 IPSec ESP 暗号化を有効にします:

switch(config-router-vlink)# **encryption ipsec spi** *spi_id* **esp** *encrypt_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *enc_keychain_name* | **null**] **authentication** *auth_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *auth_keychain_name* | **null**]

spi_id を使用してセキュリティポリシーインデックスを指定し、*encrypt_algorithm* を使用して暗号化アルゴリズムを定義できます。3DES、AES 128、または null を指定できます。番号 0、3、および 7 は、*key* の形式を指定します。認証アルゴリズムは、*auth_algorithm*（SHA-1 または NULL）で定義できます。

key-chain オプションを使用して、キーとアルゴリズムも構成できます。

Step 7 （任意）OSPFv3 情報を表示します。

switch(config)# **show running-config ospfv3**

設定例

次に、仮想リンクを暗号化する例を示します。

```
switch(config)# feature ospfv3
switch(config)# feature imp
switch(config-if)# router ospfv3 1
switch(config-router)# area 0.0.0.1 virtual-link 3.3.3.3
switch(config-router-vlink)# encryption ipsec spi ?
<256-4294967295> SPI Value
switch(config-router-vlink)# encryption ipsec spi 256 esp ?
3des Use the triple DES algorithim
aes Use the AES algorithim
key-chain  Encryption password key-chain
null Use NULL authentication
switch(config-router-vlink)# encryption ipsec spi 256 esp aes 128
123456789A123456789B123456789C12 authentication ?
null Use NULL authentication
sha1 Use the SHA1 algorithim
switch(config-router-vlink)# encryption ipsec spi 256 esp aes 128
123456789A123456789B123456789C12 authentication null
```

（注） 複数の OSPFv3 ネイバーに IPsec ESP を許可するには、次のポリシーマップをコントロール プレーンに適用する必要があります。

```
ipv6 access-list copp-acl-ipsec
10 permit ahp any any
20 permit esp any any

class-map type control-plane match-any copp-class-critical-customized-copp
match access-group name copp-acl-ipsec
policy-map type control-plane customized-copp
class copp-class-critical-customized-copp
police cir 36000 kbps bc 1280000 bytes conform transmit violate drop
control-plane
service-policy input customized-copp
```

ルータ レベルで OSPFv3 認証の構成

次のコマンドを使用して、ルータ レベルで OSPFv3 パケットを認証するように OSPFv3 ESP を構成できます。

キーチェーンの構成方法に関する詳細は、『**Cisco Nexus 9000 シリーズ NX-OS セキュリティ構成ガイド**』の「キーチェーン管理の構成」を参照してください。

始める前に

OSPFv3 が有効になっていることを確認してください。詳細は「OSPFv3 を有効にする」を参照してください。

手順の概要

1. configure terminal
2. **feature ospfv3**
3. **feature imp**
4. **router ospfv3 instance-tag**
5. [**no**] **authentication** {**ipsec spi** *spi_id* [*auth_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *auth_keychain_name* | **null**]
6. （任意） **show running-config ospfv3**
7. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	configure terminal
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**feature ospfv3** 例： `switch(config)# feature ospfv3`	OSPFv3 を有効にします。
Step 3	**feature imp** 例： `switch(config)# feature imp`	認証モードを有効にします。

Step 4	**router ospfv3 instance-tag**
	例：
	`switch(config)# router ospfv3 100`
	`switch(config-router)#`
		新規 OSPFv3 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

	コマンドまたはアクション	目的

Step 5	[**no**] **authentication** {**ipsec spi** *spi_id* [*auth_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *auth_keychain_name* | **null**]
	例：
	認証アルゴリズムおよびキー オプションの場合：
	`switch(config-router)# authentication ipsec spi 475 md5 11111111111111112222222222222222`
	キーチェーンの場合：
	`switch(config-router)# authentication ipsec spi 333 key-chain test1`
		プロセス（または VRF）レベルで OSPFv3 IPSec 認証を設定します。
		spi 引数は、セキュリティ パラメータ インデックス（SPI）を指定します。指定できる範囲は 256 ～ 4294967295 です。
		auth 引数は、認証のタイプを指定します。サポートされる値は md5 または sha1 です。
		0 の場合は、パスワードをクリア テキストで設定します。3 の場合は、パス キーを 3DES 暗号化として設定します。7 パス キーを Cisco タイプ 7 暗号化として設定します。
		cleartext オプション（0）を使用する場合、key 引数は md5 では 32 文字、sha1 では 40 文字にする必要があります。
		Cisco NX-OS リリース 10.4(1)F 以降では、**key-chain** オプションはキーおよびアルゴリズムを構成するために提供されます。
		このコマンドの **no** 形式を使用して、OSPFv3 IPSec 認証を無効にします。

Step 6	（任意） **show running-config ospfv3** 例： `switch(config)# show running-config ospfv3`	OSPFv3 認証構成情報を表示します。
Step 7	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

エリア レベルで OSPFv3 認証の構成

次のコマンドを使用して、エリア レベルで OSPFv3 パケットを認証するように OSPFv3 ESP を構成できます。

キーチェーンの構成方法に関する詳細は、『**Cisco Nexus 9000 シリーズ NX-OS セキュリティ構成ガイド**』の「キーチェーン管理の構成」を参照してください。

始める前に

OSPFv3 が有効になっていることを確認してください。詳細は「OSPFv3 を有効にする」を参照してください。

手順の概要

1. **configure terminal**
2. **feature ospfv3**
3. **feature imp**
4. **router ospfv3 instance-tag**
5. [**no**] **area** *area-id-ip* **authentication** {**ipsec spi** *spi_id*[*auth_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *auth_keychain_name* | **null**]
6. （任意） **show running-config ospfv3**
7. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**feature ospfv3** 例： `switch(config)# feature ospfv3`	OSPFv3 を有効にします。
Step 3	**feature imp** 例： `switch(config)# feature imp`	認証モードを有効にします。

Step 4	**router ospfv3 instance-tag**
	例：
	`switch(config)# router ospfv3 100`
	`switch(config-router)#`
		新規 OSPFv3 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

Step 5	[**no**] **area** *area-id-ip* **authentication** {**ipsec spi** *spi_id*[*auth_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *auth_keychain_name* | **null**]
	例：
	認証アルゴリズムおよびキー オプションの場合：
	`switch(config-router)# area 0 authentication ipsec spi 475 md5 11111111111111112222222222222222`
	キーチェーンの場合：
	`switch(config-router)# area 0 authentication ipsec spi 333 key-chain test1`
		エリア レベルで OSPFv3 IPSec 認証を設定します。
		spi 引数は、セキュリティ パラメータ インデックス（SPI）を指定します。指定できる範囲は 256 ～ 4294967295 です。
		auth 引数は、認証のタイプを指定します。サポートされる値は MD5 または SHA-1 です。
		0 の場合は、パスワードをクリア テキストで設定します。3 の場合は、パス キーを 3DES 暗号化として設定します。7：Cisco タイプ 7 暗号化としてキーを構成します。

	コマンドまたはアクション	目的
		cleartext オプション（0）を使用する場合、key 引数は MD5 では 32 文字、SHA-1 では 40 文字にする必要があります。 Cisco NX-OS リリース 10.4(1)F 以降では、**key-chain** オプションはキーおよびアルゴリズムを構成するために提供されます。 このコマンドの **no** 形式を使用して、OSPFv3 IPSec 認証を無効にします。
Step 6	（任意） **show running-config ospfv3** 例： `switch(config)# show running-config ospfv3`	OSPFv3 認証構成情報を表示します。
Step 7	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

インターフェイス レベルで OSPFv3 認証の構成

次のコマンドを使用して、間隔レベルで OSPFv3 パケットを認証するように OSPFv3 ESP を構成できます。

キーチェーンの構成方法に関する詳細は、『**Cisco Nexus 9000 シリーズ NX-OS セキュリティ構成ガイド**』の「キーチェーン管理の構成」を参照してください。

始める前に

OSPFv3 が有効になっていることを確認してください。詳細は「OSPFv3 を有効にする」を参照してください。

手順の概要

1. configure terminal
2. **interface***interface-type slot/port*
3. [**no**] **ospfv3 authentication** {**disable** | **ipsec spi** *spi_id* {**md5** *akey* | **sha1** *akey* | **key-chain** *keychain_ah*}}
4. （任意） **show running-config ospfv3**
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	configure terminal
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**interface***interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	[**no**] **ospfv3 authentication** {**disable** | **ipsec spi** *spi_id* {**md5** *akey* | **sha1** *akey* | **key-chain** *keychain_ah*}}
	例：
	認証アルゴリズムおよびキー オプションの場合：
	`switch(config-if)# ospfv3 authentication ipsec spi 475 md5 11111111111111112222222222222222`
	キーチェーンの場合：
	`switch(config-if)# ospfv3 authentication ipsec spi 333 key-chain test1`
		指定したインターフェイスの OSPFv3 IPSec 認証を設定します。
		spi 引数は、セキュリティ パラメータ インデックス（SPI）を指定します。指定できる範囲は 256 ～ 4294967295 です。
		auth 引数は、認証のタイプを指定します。サポートされる値は MD5 または SHA-1 です。
		0 の場合は、パスワードをクリア テキストで設定します。3 の場合は、パス キーを 3DES 暗号化として設定します。7：Cisco タイプ 7 暗号化としてキーを構成します。
		cleartext オプション（0）を使用する場合、key 引数は MD5 では 32 文字、SHA-1 では 40 文字にする必要があります。
		Cisco NX-OS リリース 10.4(1)F 以降では、**key-chain** オプションはキーおよびアルゴリズムを構成するために提供されます。
		このコマンドの **no** 形式を使用して、OSPFv3 IPSec 認証を無効にします。

Step 4	（任意） **show running-config ospfv3** 例： `switch(config)# show running-config ospfv3`	OSPFv3 認証構成情報を表示します。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

仮想リンク レベルでの OSPFv3 認証の構成

次のコマンドを使用して、仮想リンク レベルで OSPFv3 パケットを認証するように OSPFv3 ESP を構成できます。

キーチェーンの構成方法に関する詳細は、『**Cisco Nexus 9000 シリーズ NX-OS セキュリティ構成ガイド**』の「キーチェーン管理の構成」を参照してください。

始める前に

OSPFv3 が有効になっていることを確認してください。詳細は「OSPFv3 を有効にする」を参照してください。

手順の概要

1. configure terminal
2. **feature ospfv3**
3. **feature imp**
4. **router ospfv3 instance-tag**
5. **area** *area-id* **virtual-link** *router-id*
6. [**no**] **authentication** {**ipsec spi** *spi_id* [*auth_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *auth_keychain_name* | **null**]
7. （任意） **show running-config ospfv3**
8. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	configure terminal
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**feature ospfv3** 例： `switch(config)# feature ospfv3`	OSPFv3 を有効にします。
Step 3	**feature imp** 例： `switch(config)# feature imp`	認証モードを有効にします。
Step 4	**router ospfv3 instance-tag** 例：	新規 OSPFv3 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

	コマンドまたはアクション	目的
	`switch(config)# router ospfv3 100` `switch(config-router)#`	

Step 5	**area** *area-id* **virtual-link** *router-id*
	例：
	`switch(config-router)# area 0.0.0.10 virtual-link 2001:0DB8::1`
	`switch(config-router-vlink)#`
		リモート ルータへの仮想リンクの端を作成します。仮想リンクをリモート ルータ上に作成して、リンクを完成させる必要があります。

Step 6	[**no**] **authentication** {**ipsec spi** *spi_id* [*auth_algorithm* [**0** | **3** | **7**] *key* | **key-chain** *auth_keychain_name* | **null**]
	例：
	認証アルゴリズムおよびキー オプションの場合：
	`switch(config-router-vlink)# authentication ipsec spi 475 md5 11111111111111112222222222222222`
	キーチェーンの場合：
	`switch(config-router-vlink)# authentication ipsec spi 333 key-chain test1`
		仮想リンク レベルで OSPFv3 IPSec 認証を構成します。
		spi 引数は、セキュリティ パラメータ インデックス（SPI）を指定します。指定できる範囲は 256 ～ 4294967295 です。
		auth 引数は、認証のタイプを指定します。サポートされる値は MD5 または SHA-1 です。
		0 の場合は、パスワードをクリア テキストで設定します。3 の場合は、パス キーを 3DES 暗号化として設定します。7：Cisco タイプ 7 暗号化としてキーを構成します。
		cleartext オプション（0）を使用する場合、key 引数は MD5 では 32 文字、SHA-1 では 40 文字にする必要があります。
		Cisco NX-OS リリース 10.4(1)F 以降では、**key-chain** オプションはキーおよびアルゴリズムを構成するために提供されます。
		このコマンドの **no** 形式を使用して、OSPFv3 IPSec 認証を無効にします。

Step 7	（任意） **show running-config ospfv3** 例： `switch(config)# show running-config ospfv3`	OSPFv3 認証構成情報を表示します。
Step 8	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

OSPFv3 の設定の確認

OSPFv3 の設定を表示するには、次のいずれかのタスクを実行します。

コマンド	目的
show ipv6 ospfv3 [instance-tag] [vrf vrf-name]	1 つまたは複数の OSPFv3 ルーティング インスタンスに関する情報が表示されます。出力には、次のエリア レベルのカウントが含まれます。 • このエリアのインターフェイス：このエリアに追加されたすべてのインターフェイスの数（設定されたインターフェイス）。 • アクティブ インターフェイス：ルータリンクステートおよびSPF（UPインターフェイス）にあると見なされるすべてのインターフェイスの数。 • パッシブ インターフェイス：OSPF パッシブと見なされるすべてのインターフェイスの数（隣接関係は形成されません）。 • ループバックインターフェイス：すべてのローカルループバックインターフェイスの数。
show ipv6 ospfv3 border-routers	ABR および ASBR への内部 OSPF ルーティング テーブル エントリを表示します。
show ipv6 ospfv3 database	特定のルータの OSPFv3 データベースに関する情報のリストを表示します。
show ipv6 ospfv3 interface *type number* [**vrf** {*vrf-name* | **all** | **default** | **management**}]	OSPFv3 インターフェイス情報を表示します。
show ipv6 ospfv3 neighbors	ネイバー情報を表示します。**clear ospfv3 neighbors** コマンドを使用すると、すべてのネイバーとの隣接関係を削除できます。
show ipv6 ospfv3 request-list	ルータから要求されている LSA の一覧を表示します。
show ipv6 ospfv3 retransmission-list	再送を待っている LSA の一覧を表示します。
show ipv6 ospfv3 summary-address	OSPFv3 インスタンスで設定されている、すべての集約アドレス再配布情報の一覧を表示します。
show ospfv3 process	プロセス レベルの OSPFv3 認証設定を表示します。

コマンド	目的
show ospfv3 interface *interface-type slot/port*	インターフェイス レベルでの OSPFv3 認証設定を表示します。
show running-configuration ospfv3	現在実行中の OSPFv3 コンフィギュレーションを表示します。

OSPFv3のモニタリング

OSPFv3 統計情報を表示するには、次のコマンドを使用します。

コマンド	目的
show ipv6 ospfv3 memory	OSPFv3 メモリ使用状況の統計情報を表示します。
show ipv6 ospfv3 policy statistics area *area-id* **filter-list** {**in** | **out**} [**vrf** {*vrf-name* | **all** | **default** | **management**}]	エリアの OSPFv3 ルート ポリシー統計情報を表示します。
show ipv6 ospfv3 policy statistics redistribute {**bgp** *id* | **direct** | **isis** *id* | **rip** *id* | **static vrf** {*vrf-name* | **all** | **default** | **management**}]	OSPFv3 ルート ポリシー統計を表示します。
show ipv6 ospfv3 statistics [**vrf** {*vrf-name* | **all** | **default** | **management**}]	OSPFv3 イベント カウンタを表示します。
show ipv6 ospfv3 traffic *interface-type number* [**vrf** {*vrf-name* | **all** | **default** | **management**}]	OSPFv3 パケット カウンタを表示します。

OSPFv3 の設定例

次に、OSPFv3 を設定する例を示します。

```
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
```

key-chain オプションを使用して、OSPFv3 暗号を構成する例を示します。

```
switch(config-if)# ospfv3 encryption ipsec spi 333 esp ?
  3des       Use the triple DES algorithim
  aes        Use the AES algorithim
  key-chain  Encryption password key-chain
  null       Use NULL authentication
switch(config-if)# ospfv3 encryption ipsec spi 333 esp key-chain ?
  WORD  Encryption key-chain name (Max Size 63)
```

```
switch(config-if)# ospfv3 encryption ipsec spi 333 esp key-chain test1 ?
  authentication  Specify authentication parameters
switch(config-if)# ospfv3 encryption ipsec spi 333 esp key-chain test1 authentication ?
  key-chain  Authentication password key-chain
  null       Use NULL authentication
switch(config-if)# ospfv3 encryption ipsec spi 333 esp key-chain test1 authentication
key-chain ?
  WORD Authentication key-chain name (Max Size 63)
switch(config-if)# ospfv3 encryption ipsec spi 333 esp key-chain test1 authentication
key-chain test2 ?
  <CR>
switch(config-router)# sh ospfv3
 Routing Process 2 with ID 20.20.10.2 VRF default
 Routing Process Instance Number 1
 Install discard route for summarized internal routes.
 ESP Encryption 3DES, Authentication SHA1, SPI 334, ConnId 334
 ESP keychains: Encr test_key_chain_01(ready), Auth test1(ready)
 Number of new LSAs originated : 3
 Number of new LSAs received : 0
```

関連項目

次の項目には、OSPF に関する詳細情報が含まれています。

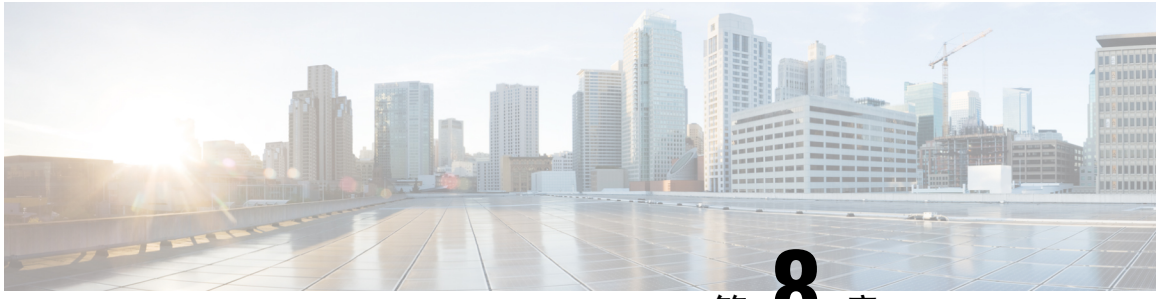
- OSPFv2 の設定 （113 ページ）
- Route Policy Manager の設定 （587 ページ）

その他の参考資料

OSPF の実装に関する詳細情報については、次のページを参照してください。

MIB

MIB	MIB のリンク
OSPFv3 に関連する MIB	サポートされている MIB を検索およびダウンロードするには、次の URL にアクセスしてください。 ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html



第 8 章

EIGRP の設定

この章では、Cisco NX-OS デバイスで Enhanced Interior Gateway Routing Protocol（EIGRP）を設定する方法について説明します。

- EIGRP について （237 ページ）
- EIGRP の前提条件 （246 ページ）
- EIGRP の注意事項と制約事項 （246 ページ）
- デフォルト設定 （248 ページ）
- 基本的な EIGRP の設定 （249 ページ）
- 高度な EIGRP の設定 （255 ページ）
- EIGRP の仮想化の設定 （273 ページ）
- EIGRP の設定の確認 （274 ページ）
- EIGRP のモニタリング （275 ページ）
- EIGRP の設定例 （275 ページ）
- 関連項目 （276 ページ）
- その他の参考資料 （276 ページ）

EIGRP について

EIGRP は、リンクステート プロトコルの機能にディスタンス ベクトル プロトコルの利点を組み合わせたプロトコルです。EIGRP は、定期的に Hello メッセージを送信してネイバーを探索します。EIGRP は、新規ネイバーを検出すると、すべてのローカル EIGRP ルートおよびルート メトリックに対する 1 回限りの更新を送信します。受信側の EIGRP ルータは、受信したメトリックと、その新規ネイバーにローカルで割り当てられたリンクのコストに基づいて、ルート ディスタンスを計算します。この最初の全面的なルート テーブルの更新後は、ルート変更の影響を受けるネイバーにのみ、差分更新が EIGRP により送信されます。この処理により、コンバージェンスにかかる時間が短縮され、EIGRP が使用する帯域幅が最小限になります。

EIGRP コンポーネント

EIGRP には、次の基本コンポーネントがあります。

- Reliable Transport Protocol
- ネイバー探索およびネイバー回復
- 拡散更新アルゴリズム

信頼性の高いトランスポート プロトコル

信頼性の高いトランスポートプロトコルは、すべてのネイバーにEIGRPパケットの順序付けされた配信を保証します。(「ネイバー探索およびネイバー回復」の項を参照してください。) 信頼性の高いトランスポート プロトコルは、マルチキャスト パケットとユニキャスト パケットの混合伝送をサポートしています。この転送は信頼性が高く、未確認パケットが保留されているときにも、マルチキャスト パケットの迅速な送信が可能です。この方式により、さまざまな速度のリンクでも短いコンバージェンス時間が維持されるようになります。マルチキャスト パケットとユニキャストパケットの送信を制御するデフォルトタイマーの変更の詳細については、高度なEIGRPの設定 （255 ページ） を参照してください。

Reliable Transport Protocol には、次のメッセージ タイプが含まれます。

- Hello：ネイバー探索およびネイバー回復に使用されます。EIGRPはデフォルトでは、定期的なマルチキャスト Hello メッセージをローカル ネットワーク上に、設定された hello 間隔で送信します。デフォルトの hello 間隔は 5 秒です。
- 確認：更新、照会、返信を確実に受信したことを確認します。
- 更新：ルーティング情報が変更されると、その影響を受けるネイバーに送信されます。更新には、ルートの宛先、アドレスマスク、および遅延や帯域幅などのルートメトリックが含まれます。更新情報は EIGRP トポロジ テーブルに格納されます。
- 照会および返信：EIGRP が使用する拡散更新アルゴリズムの一部として送信されます。

ネイバー探索およびネイバー回復

EIGRP は、Reliable Transport Protocol からの Hello メッセージを使用して、直接接続されたネットワーク上のネイバー EIGRP ルータを探索します。EIGRP により、ネイバー テーブルにネイバーが追加されます。ネイバーテーブルの情報には、ネイバーアドレス、検出されたインターフェイス、およびネイバー到達不能を宣言する前にEIGRPが待機する時間を示すホールドタイムが含まれています。デフォルトのホールド タイムは、hello 間隔の 3 倍または 15 秒です。

EIGRPは、ローカルEIGRPルーティング情報を共有するために、一連の更新メッセージを新規ネイバーに送信します。このルート情報は EIGRP トポロジ テーブルに格納されます。このようにEIGRP ルート情報全体を最初に送信した後は、ルーティングが変更されたときにのみ、EIGRP により更新メッセージが送信されます。これらの更新メッセージは新情報または更新情報のみを含んでおり、変更の影響を受けるネイバーにのみ送信されます。「EIGRP ルート アップデート」の項を参照してください。

EIGRP はネイバーへのキープアライブとして、Hello メッセージも使用します。Hello メッセージを受信している限り、Cisco NX-OS は、ネイバーがダウンせずに機能していると判定します。

拡散更新アルゴリズム

拡散更新アルゴリズム（DUAL）により、トポロジ テーブルの宛先ネットワークに基づいてルーティング情報が計算されます。トポロジ テーブルには、次の情報が含まれます。

- IPv4 または IPv6 アドレス/マスク：この宛先のマスクのネットワーク アドレスおよびネットワーク マスク。
- サクセサ：現在のフィジブル ディスタンスよりも宛先まで短いディスタンスをアドバタイズする、すべてのフィジブル サクセサまたはネイバーの IP アドレスおよびローカル インターフェイス接続。
- フィージビリティ ディスタンス（FD）：計算された、宛先までの最短ディスタンス。

DUAL は、ディスタンス メトリックを使用して、ループが発生しない効率的なパスを選択します。DUAL はルートを選択し、フィジブル サクセサに基づいてユニキャスト ルーティング情報ベース（RIB）に挿入します。トポロジが変更されると、DUAL は、トポロジ テーブルでフィジブル サクセサを探します。フィジブル サクセサが見つかった場合、DUAL は、最短のフィジブル ディスタンスを持つフィジブル サクセサを選択して、それをユニキャスト RIB に挿入します。これにより、再計算が不要となります。

フィジブル サクセサが存在しないが、宛先をアドバタイズするネイバーが存在する場合は、DUAL がパッシブ状態からアクティブ状態へと移行し、新しいサクセサまたは宛先へのネクスト ホップ ルータを決定する再計算をトリガーします。ルートの再計算に必要な時間は、コンバージェンス時間に影響します。EIGRP は照会メッセージをすべてのネイバーに送信し、フィジブル サクセサを探します。フィジブル サクセサを持つネイバーは、その情報を含む返信メッセージを送信します。フィジブル サクセサを持たないネイバーは、DUAL の再計算をトリガーします。

EIGRP ルート更新

トポロジが変更されると、EIGRP は、変更されたルーティング情報のみを含む更新メッセージを影響を受けるネイバーに送信します。更新メッセージには、新規の、または更新されたネットワーク宛先へのディスタンス情報が含まれます。

EIGRP でのディスタンス情報は、帯域幅、遅延、負荷使用状況、リンクの信頼性などの使用可能なルート メトリックの組み合わせとして表現されます。各メトリックには重みが関連付けられており、これにより、メトリックがディスタンスの計算に含まれるかどうかが決定します。このメトリックの重みは設定することができます。特性を微調整して最適なパスを完成することもできますが、設定可能なメトリックの大部分でデフォルト設定を使用することを推奨します。

内部ルート メトリック

内部ルートとは、同じ EIGRP 自律システム内のネイバー間のルートです。これらのルートには、次のメトリックがあります。

- ネクスト ホップ：ネクスト ホップ ルータの IP アドレス。
- 遅延：宛先ネットワークへのルートを形成するインターフェイス上で設定された遅延の合計。遅延は 10 マイクロ秒単位で設定されます。

- 帯域幅：宛先へのルートの一部であるインターフェイスで設定された最小帯域幅から計算されます。

（注） Cisco ではデフォルト帯域幅の値の使用を推奨します。この帯域幅パラメータは EIGRP でも使用されます。

- MTU：宛先へのルート上の最大伝送単位の最小値。
- ホップ カウント：宛先までにルートが通過するホップまたはルータの数。このメトリックは、DUAL 計算で直接には使用されません。
- 信頼性：宛先までのリンクの信頼性を示します。
- 負荷：宛先までのリンク上のトラフィック量を示します。

デフォルトで EIGRP は、帯域幅と遅延のメトリックを使用して、宛先までのディスタンスを計算します。計算に他のメトリックが含まれるように、メトリックの重みを変更できます。

ワイド メトリックス

EIGRP は、より高速なインターフェイスまたはバンドルされたインターフェイス上でのルート選択を改善するためのワイド（64 ビット）メトリックをサポートします。ワイド メトリックをサポートしているルータは、次のように、ワイド メトリックをサポートしていないルータと相互運用できます。

- ワイド メトリックをサポートするルータ：ローカル ワイド メトリック値を受信した値に追加し、情報を送信します。
- ワイド メトリックをサポートしないルータ：値を変更せずに受信したメトリックを送信します。

EIGRP は、ワイド メトリックのパス コストを計算するために、次の式を使用します。

メトリック = [k1 × 帯域幅 +（k2 × 帯域幅）/（256 –負荷）+ k3 × 遅延 + k6 × 拡張属性] x [k5/（信頼性 + k4）]

ユニキャスト RIB が 64 ビットのメトリック値をサポートできないため、EIGRP ワイド メトリックは RIB スケール係数で次の式を使用して、64 ビット メトリック値を 32 ビット値に変換します。

RIB メトリック =（ワイド メトリック/RIB スケール値）

RIB スケール値は設定可能なパラメータです。

EIGRP ワイドメトリックは、EIGRP メトリックの設定の k6 として、次の 2 種類の新しいメトリック値を導入します。

- ジッタ：（マイクロ秒単位で測定）ルート パス上のすべてのリンクにわたって累積します。
- エネルギー：（キロ ビット単位のワットで測定）ルートパス上のすべてのリンクにわたって累積します。

EIGRP は、ジッターやエネルギー メトリック値を持たないパス、またはより低いジッターやエネルギー メトリック値を持つパスを、より高い値のパスを持つパスよりも優先します。

（注） EIGRP ワイド メトリックは、TLV バージョン 2 で送信されます。詳細については、「ワイドメトリックの有効化」の項を参照してください。

外部ルート メトリック

外部ルートとは、異なる EIGRP 自律システムにあるネイバー間のルートです。これらのルートには、次のメトリックがあります。

- ネクスト ホップ：ネクスト ホップ ルータの IP アドレス。
- ルータ ID：このルートを EIGRP に再配布したルータのルータ ID。
- 自律システム番号：宛先の自律システム番号。
- プロトコル ID：宛先へのルートを学習したルーティング プロトコルを表すコード。
- タグ：ルート マップで使用可能な任意のタグ。
- メトリック：外部ルーティング プロトコルの、このルートのルート メトリック。

EIGRP とユニキャスト RIB

EIGRP は、すべての学習したルートを EIGRP トポロジ テーブルとユニキャスト RIB に追加します。トポロジが変更されると、EIGRP は、これらのルートを使用してフィジブル サクセサを探します。EIGRP は、他のルーティング プロトコルから EIGRP に再配布されたあらゆるルートの変更についてのユニキャスト RIB からの通知も待ち受けます。

高度な EIGRP

EIGRP の高度な機能を使用して、EIGRP の設定を最適化できます。

アドレス ファミリ

EIGRP では、IPv4 と IPv6 の両方のアドレス ファミリをサポートしています。下位互換性を保つために、ルート コンフィギュレーション モードまたは IPv4 アドレス ファミリ モードで EIGRPv4 を設定できます。アドレス ファミリ モードで IPv6 の EIGRP を設定する必要があります。

アドレス ファミリ コンフィギュレーション モードには、次の EIGRP 機能が含まれます。

- 認証
- AS 番号
- デフォルト ルート
- メトリック

- ディスタンス
- グレースフル リスタート
- ロギング
- ロード バランシング
- 再分配
- ルータ ID
- スタブ ルータ
- タイマー

複数のコンフィギュレーション モードで同じ機能を設定できません。たとえばルータ コンフィギュレーションモードでデフォルトメトリックを設定すると、アドレスファミリモードでデフォルト メトリックを設定できません。

認証

EIGRP メッセージに認証を設定することで、ネットワークでの不正なルーティング更新や無効なルーティング更新を防止できます。EIGRP認証はMD5認証ダイジェストをサポートしています。

認証キーのキーチェーン管理を使用して、仮想ルーティング/転送（VRF）インスタンスごと、またはインターフェイスごとにEIGRP認証を設定できます。キーチェーン管理を使用すると、MD5認証ダイジェストが使用する認証キーへの変更を管理できます。キー チェーンの作成の詳細については、『*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*』を参照してください。

MD5 認証を行うには、ローカル ルータとすべてのリモート EIGRP ネイバーで同一のパスワードを設定します。EIGRP メッセージが作成されると、Cisco NX-OS は、そのメッセージ自体と暗号化されたパスワードに基づいて MD5 一方向メッセージ ダイジェストを作成し、このダイジェストを EIGRP メッセージとともに送信します。受信する EIGRP ネイバーは、同じ暗号化パスワードを使用して、このダイジェストを確認します。メッセージが変更されていない場合は計算が同一であるため、EIGRP メッセージは有効と見なされます。

MD5 認証には各 EIGRP メッセージのシーケンス番号も含まれており、これにより、ネットワークでのメッセージの再送が防止されます。

スタブ ルータ

EIGRP スタブ ルーティング機能を使用すると、ネットワークの安定性の向上、リソース使用量の削減、スタブ ルータ設定の簡易化を実現できます。スタブ ルータは、リモート ルータ経由で EIGRP ネットワークに接続します。「スタブ ルーティング」の項を参照してください。

EIGRP スタブ ルーティングを使用すると、EIGRP を使用するように配布とリモート ルータを設定し、リモートルータのみをスタブとして設定する必要があります。EIGRPスタブルーティングで、分散ルータでの集約が自動的にイネーブルになるわけではありません。ほとんどの場合、分散ルータでの集約の設定が必要です。

EIGRP スタブ ルーティングを使用しない場合は、分散ルータからリモート ルータに送信されたルートがフィルタリングまたは集約された後でも、問題が発生することがあります。たとえば、ルートが企業ネットワーク内のどこかで失われた場合に、EIGRP が分散ルータに照会を送信することがあります。分散ルータは、ルートが集約されている場合でも、リモート ルータに照会を送信することがあります。分散ルータとリモート ルータの間の WAN リンク上の通信で問題が発生した場合はEIGRPがアクティブ状態のままとなり、ネットワークの他の場所が不安定となる場合があります。EIGRPスタブルーティングを使用すると、リモートルータに照会が送信されなくなります。

ルート集約

指定したインターフェイスにサマリー集約アドレスを設定できます。ルート集約を使用すると、固有性の強い一連のアドレスをすべての固有アドレスを代表する 1 つのアドレスに置き換えることによって、ルート テーブルを簡素化できます。たとえば、10.1.1.0/24、10.1.2.0/24、および 10.1.3.0/24 というアドレスを 1 つの集約アドレス 10.1.0.0/16 に置き換えることができます。

より具体的なアドレスがルーティング テーブルにある場合、EIGRP は、より具体的なルートの最小メトリックに等しいメトリックを持つインターフェイスからの集約アドレスをアドバタイズします。

プロセスの再起動またはシステムスイッチオーバーの場合、サマリーアドレスによってトラフィックが失われる可能性があります。トラフィックは、サマリーアドレスを使用してトラフィックがルーティングされる PEER で確認されます。

（注） EIGRP は、自動ルート集約をサポートしていません。

ルートの再配布

EIGRP を使用すると、スタティック ルート、他の EIGRP AS が学習したルート、またはほかのプロトコルからのルートを再配布できます。再配布を指定したルート マップを設定して、どのルートがEIGRPに渡されるかを制御する必要があります。ルートマップを使用すると、宛先、送信元プロトコル、ルートタイプ、ルートタグなどの属性に基づいて、ルートをフィルタリングできます。Route Policy Manager の設定 （587 ページ）を参照してください。

インポートされた EIGRP へのすべてのルートに使用されるデフォルト メトリックも設定できます。

ルーティング アップデートからルートをフィルタリングするには、配布リストを使用します。これらのフィルタ処理されたルートは、**ip distribute-list eigrp** コマンドで各インターフェイスに適用されます。

ロード バランシング

ロード バランシングを使用すると、ルータは、宛先アドレスから等距離内にあるすべてのルータのネットワーク ポートにトラフィックを分散できます。ロードバランシングにより、ネットワーク セグメントの使用率が向上し、それによってネットワーク帯域幅の効率も向上します。

Cisco NX-OS は、EIGRP ルート テーブルおよびユニキャスト RIB 中の 16 までの等コスト パスを使用する等コスト マルチパス（ECMP）機能をサポートしています。これらのパスの一部または全部に対してトラフィックのロード バランスを行うよう、EIGRP を設定できます。

（注） Cisco NX-OS の EIGRP は、等コストでないロード バランシングをサポートしていません。

Split Horizon

スプリットホライズンを使用すると、ルートを学習したインターフェイスからEIGRPがルートをアドバタイズしないようにできます。

スプリット ホライズンは、EIGRP 更新パケットおよび EIGRP 照会パケットの送信を制御する方式です。インターフェイスでスプリット ホライズンをイネーブルにすると、Cisco NX-OS は、このインターフェイスから学習された宛先への更新パケットも照会パケットも送信しません。この方法でアップデート パケットとクエリー パケットを制御すると、ルーティング ループが発生する可能性が低くなります。

EIGRP はポイズン リバースによるスプリット ホライズンにより、EIGRP がルートを学習したインターフェイス経由で、そのルートを到達不能としてアドバタイズするよう設定されます。

EIGRP は、次のシナリオでスプリット ホライズン、またはポイズン リバースによるスプリット ホライズンを使用します。

- スタートアップ モードで、2 台のルータ間で初めてトポロジ テーブルを交換する。
- トポロジ テーブルの変更をアドバタイズする。
- 照会メッセージを送信する。

デフォルトでは、スプリット ホライズン機能がすべてのインターフェイスでイネーブルになっています。

BFD

この機能では、IPv4 および IPv6 用の双方向フォワーディング検出（BFD）をサポートします。BFD は、転送パスの障害を高速で検出することを目的にした検出プロトコルです。BFD は 2 台の隣接デバイス間のサブセカンド障害を検出し、BFD の負荷の一部を、サポートされるモジュール上のデータ プレーンに分散できるため、プロトコル hello メッセージよりも CPU を使いません。詳細については、『Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide』を参照してください。

仮想化のサポート

EIGRP は、仮想ルーティングおよび転送（VRF）インスタンスをサポートしています。

グレースフル リスタートおよびハイ アベイラビリティ

Cisco NX-OS は、EIGRP の無停止フォワーディングおよびグレースフル リスタートをサポートします。

EIGRP の NSF を使用すると、フェールオーバー後に EIGRP ルーティング プロトコル情報が復元される間に、データ パケットを FIB 内の既存のルートで転送できます。ノンストップ フォワーディング（NSF）を使用すると、ピア ネットワーキングデバイスでルーティング フラップが発生することがありません。フェールオーバー時に、データ トラフィックはインテリジェントモジュール経由で転送され、スタンバイ スーパーバイザがアクティブになります。

Cisco NX-OS システムでコールド リブートが発生した場合、デバイスはシステムへのトラフィック転送を中止し、ネットワーク トポロジからシステムを削除します。このシナリオでは、EIGRP でステートレス再起動が発生し、すべてのネイバーが削除されます。Cisco NX-OS はスタートアップ構成を適用し、EIGRP がネイバーを再検出して、完全な EIGRP ルーティング情報を再度共有します。

Cisco NX-OS を実行するデュアル スーパーバイザ プラットフォームで、ステートフル スーパーバイザ スイッチオーバーが発生します。このスイッチオーバーが発生する前に、EIGRP はグレースフル リスタートを使用して、EIGRP がしばらく使用不可であることを宣言します。スイッチオーバーの間、EIGRP は無停止フォワーディングを使用して FIB の情報に基づいてトラフィックを転送し続け、システムがネットワーク トポロジから取り除かれることはありません。

グレースフル リスタート対応ルータは、Hello メッセージを使用して、グレースフル リスタート動作が開始されたことをネイバーに通知します。グレースフル リスタート認識ルータが、グレースフル リスタート対応ネイバーからグレースフル リスタート動作が進行中であるという通知を受信すると、両方のルータは各トポロジテーブルをただちに交換します。グレースフル リスタート認識ルータは、ルータの再起動を支援するための次のアクションを実行します。

- ルータは、EIGRP Hello 保持時間を失効し、Hello メッセージにセットされる間隔を短くします。このプロセスにより、グレースフル リスタート認識ルータは再起動中のルータにより早く応答し、再起動中のルータがネイバーを再検出し、トポロジ テーブルを再構築するために必要な時間を短縮します。
- ルータは、ルート保留タイマーを開始します。このタイマーで、グレースフル リスタート認識ルータが、再起動中のネイバー ルータのために既知のルートを保留する時間の長さが設定されます。デフォルトの期間は 240 秒です。
- ルータは、ネイバーが再起動していることをピア リストに記載する、隣接関係を維持する、グレースフル リスタート認識ルータのトポロジテーブルを送信する準備ができたことを知らせるシグナルをネイバーが送信するか、ルートホールド タイマーが期限切れになるまで再起動中のネイバーを保持する、ということを行います。グレースフル リスタート認識ルータ上でルート保留タイマーの期限が切れた場合、グレースフル リスタート認識ルータは保留ルートを破棄し、再起動中のルータをネットワークに参加する新しいルータとして扱い、隣接関係を再確立します。

スイッチオーバー後に、Cisco NX-OS は実行コンフィギュレーションを適用し、EIGRP は、自身が再び稼働していることをネイバーに通知します。

複数の EIGRP インスタンス

Cisco NX-OSは、同一システム上で動作する複数の EIGRP プロトコル インスタンスをサポートします。すべてのインスタンスで同じシステム ルータ ID を使用します。インスタンスごとに一意のルータ ID を設定することもできます。サポートされる EIGRP インスタンスの数については、『Cisco Nexus 9000 Series NX-OS Verified Scalability Guide』を参照してください。

EIGRP の前提条件

EIGRP を使用するには、次の前提条件を満たしている必要があります。

- EIGRP 機能がイネーブルにする必要があります（「EIGRP 機能のいネーブル化（Enabling the EIGRP Feature）」を参照）。

EIGRP の注意事項と制約事項

EIGRP 設定時の注意事項および制約事項は次のとおりです。

- テーブルマップ、ルートのアドミニストレーティブディスタンス、およびメトリックを設定すると、コンフィギュレーションコマンドによってEIGRPネイバーがフラップします。これは予期された動作です。
- プレフィックス リスト内の名前は、大文字と小文字が区別されません。一意の名前を使用することを推奨します。大文字と小文字を変更して同じ名前を使用しないでください。たとえば、CTCPrimaryNetworks と CtcPrimaryNetworks は 2 つの異なるエントリではありません。
- 他のプロトコル、接続されたルータ、またはスタティック ルートからの再配布には、メトリック設定（デフォルト メトリック設定オプションまたはルート マップによる）が必要です。Route Policy Manager の設定 （587 ページ）を参照してください。
- グレースフル スタートについては、NSF 認識ルータが動作中であり、ネットワークで完全に収束している場合にのみ、このルータがNSF対応ルータのグレースフル リスタート動作を支援できます。
- グレースフル スタートについては、NSF 認識ルータが動作中であり、ネットワークで完全に収束している場合にのみ、このルータがNSF対応ルータのグレースフル リスタート動作を支援できます。
- グレースフル リスタートについては、グレースフル リスタートに関係する隣接デバイスが NSF 認識、または NSF 対応である必要があります。
- Cisco NX-OS EIGRP は Cisco IOS ソフトウェアの EIGRP と互換性があります。
- EIGRP はトンネル インターフェイスではサポートされていません。

- 妥当な理由がない限り、メトリックの重みを変更しないでください。メトリックの重みを変更した場合は、同じ自律システム内のすべての EIGRP ルータに、それを適用する必要があります。
- 1 ギガビット以上のインターフェイス速度の EIGRP ネットワークでの標準メトリックとワイド メトリックの組み合わせは、最適なルーティングになる可能性があります。
- 大規模ネットワークの場合は、スタブの使用を検討してください。
- NX-OS リリース 10.3(3)F までは、 EIGRP の最大プレフィックス再配布機能がデフォルトで有効になっており、デフォルトの制限は 10000（10K）です。以降のリリースでは、この構成はデフォルトで有効になっていません。NX-OS リリース 10.3(3)F からアップグレードすると、ユーザーが最大プレフィックス値を 10000 として明示的に構成していた場合でも、このデフォルト設定は削除されます。ユーザーが 10000 以外の制限を構成していた場合にのみ、10.3(3)F からのアップグレード後に構成が残ります。
- EIGRP ベクトル メトリックは維持されないため、異なる EIGRP 自律システム間での再配布は避けてください。
- **no** {**ip** | **ipv6**} **next-hop-self** コマンドは、ネクスト ホップの到達可能性を保証しません。
- {**ip** | **ipv6**} **passive-interface eigrp** コマンドを使用すると、ネイバーが形成されなくなります。
- Cisco NX-OS は IGRP も、IGRP および EIGRP クラウドの接続もサポートしていません。
- 自動集約はデフォルトで無効になっており、有効にすることはできません。
- Cisco NX-OS は IP のみをサポートしています。
- ハイ アベイラビリティは、EIGRP 集約タイマーでサポートされません。
- デフォルト以外のアグレッシブ hello タイマーを構成するには、EIGRP のデフォルト タイマーで BFD を使用することを推奨します。
- Cisco NX-OS リリース 9.3(4) 以降では、ルートを EIGRP に再配布し、ルート マップまたはプレフィックス リストを使用してプレフィックスをフィルタリングするときに、触れてない場合でもフィルタによって許可されているすべてのプレフィックスは、EIGRP トポロジテーブル内で更新されます。この更新は、このプレフィックスセットのクエリ ドメイン内のすべての EIGRP ルータに通知されます。
- Cisco NX-OS リリース 10.3(1)F 以降、EIGRP は Cisco Nexus 9808 プラットフォーム スイッチでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、EIGRP は Cisco Nexus 9804 プラットフォーム スイッチでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、EIGRP は、Cisco Nexus 9808 および 9804 スイッチを搭載した N9KX98900CD-A および N9KX9836DM-A ライン カードでサポートされます。
- ASCII リロードにより、VRF 構成は EIGRP の下のすべての VRF に対して自動的に追加されます

（注） Cisco IOS の CLI に慣れている場合、この機能の Cisco NX-OS コマンドは従来の Cisco IOS コマンドと異なる点があるため注意が必要です。

デフォルト設定

テーブルは、各 EIGRP パラメータに対するデフォルト設定を示します。

表 20 : EIGRP パラメータのデフォルト設定

パラメータ	デフォルト
アドミニストレーティブ ディスタンス	• 内部ルート：90 • 外部ルート：170
帯域幅の割合	50%
再配布されたルートのデフォルトのメトリック	• 帯域幅：100000 Kb/s • 遅延：100（10 マイクロ秒単位） • 信頼性：255 • ロード：1 • MTU：1500
EIGRP 機能	ディセーブル
hello 間隔	5 秒
Hold time	15 秒
等コスト パス	8
メトリック重み	0 1 0 1 0 0 0
アドバタイズされたネクスト ホップ アドレス	ローカル インターフェイスの IP アドレス
NSF コンバージェンス時間	120
NSF ルート保留時間	240
NSF 信号送信時間	20
再分配	ディセーブル
スプリット ホライズン	有効（Enabled）

基本的な EIGRP の設定

基本的な EIGRP の設定。

EIGRP 機能の有効化

EIGRP を設定するには、その前に EIGRP を有効にする必要があります。

手順の概要

1. **configure terminal**
2. [**no**] **feature eigrp**
3. （任意） **show feature**
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	[**no**] **feature eigrp** 例： `switch(config)# feature eigrp`	EIGRP 機能を有効にします。 **no** オプションを使用すると、EIGRP 機能が無効になり、関連する設定がすべて削除されます。
Step 3	（任意） **show feature** 例： `switch(config)# show feature`	有効にされた機能に関する情報を表示し。
Step 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

EIGRP インスタンスの作成

EIGRP インスタンスを作成して、そのインスタンスにインターフェイスを関連付けることができます。この EIGRP プロセスに一意の自律システム番号を割り当てます（「自律システム」の項を

参照）。ルート再配布をイネーブルにしていない限り、他の自律システムからルートがアドバタイズされることも、受信されることもありません。

始める前に

EIGRP 機能がイネーブルにする必要があります（「EIGRP 機能のいネーブル化（Enabling the EIGRP Feature）」を参照）。

EIGRP がルータ ID（設定済みのループバック アドレスなど）を入手可能であるか、またはルータ ID オプションを設定する必要があります。

自律システム番号であると認められていないインスタンス タグを設定する場合は、自律システム番号を明示的に設定する必要があります。そうしないと、この EIGRP インスタンスはシャットダウン状態のままになります。IPv6 の場合、この番号は、アドレス ファミリの下で設定する必要があります。

手順の概要

1. **configure terminal**
2. [**no**] **router eigrp** *instance-tag*
3. （任意） **autonomous-system** *as-number*
4. （任意） **log-adjacency-changes**
5. （任意） **log-neighbor-warnings** [*seconds*]
6. **interface** *interface-type slot/port*
7. {**ip** | **ipv6**} **router eigrp** *instance-tag*
8. （任意） **show** {**ip** | **ipv6**} **eigrp interfaces**
9. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	[**no**] **router eigrp** *instance-tag*
	例：
	`switch(config)# router eigrp Test1`
	`switch(config-router)#`
		インスタンス タグを設定して、新しい EIGRP プロセスを作成します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。
		AS 番号であると認められていない *instance-tag* を設定する場合は、**autonomous-system** コマンドを使用して AS 番号を明示的に設定する必要があります。そ

	コマンドまたはアクション	目的
		うしないと、この EIGRP インスタンスはシャットダウン状態のままになります。 **no** オプションを使用すると、EIGRP プロセスとそれに関連する設定がすべて削除されます。 （注） EIGRP プロセスを削除する場合は、インターフェイス モードで設定された EIGRP コマンドも削除する必要があります。
Step 3	（任意） **autonomous-system** *as-number* 例： `switch(config-router)# autonomous-system 33`	この EIGRP インスタンスに一意の AS 番号を設定します。有効な範囲は 1 ～ 65535 です。
Step 4	（任意） **log-adjacency-changes** 例： `switch(config-router)# log-adjacency-changes`	隣接関係の状態が変化するたびに、システムメッセージを生成します。このコマンドは、デフォルトでイネーブルになっています。
Step 5	（任意） **log-neighbor-warnings** [*seconds*] 例： `switch(config-router)# log-neighbor-warnings`	ネイバーの警告が発生するたびに、システムメッセージを生成します。警告メッセージの時間間隔を、1～65535 の秒数で設定できます。デフォルトは 10 秒です。このコマンドは、デフォルトでイネーブルになっています。

Step 6	必須: **interface** *interface-type slot/port*
	例：
	`switch(config-router)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。**?** を使用すると、スロットおよびポートの範囲を確認できます。

Step 7	必須: {**ip** | **ipv6**} **router eigrp** *instance-tag*
	例：
	`switch(config-if)# ip router eigrp Test1`
	`R2(config-if)# vrf member eigrp-vrf`
	`Warning: Retain-L3-config is on, deleted and re-added L3 config on interface Ethernet1/8`
	`VRF eigrp-vrf does not exist. Create vrf to make interface Ethernet1/8 operational`
	`R2(config-if)#`
	`R2(config-if)# sh ru eigrp`
	`!Command: show running-config eigrp`
	`!Running configuration last done at: Thu Aug 25 06:59:31 2022`
	`!Time: Thu Aug 25 06:59:36 2022`
		このインターフェイスを、設定された EIGRP プロセスに関連付けます。インスタンスタグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。
		（注）
		EIGRP プロセスが実行され、*vrf retain* が構成されているインターフェイスでは、この場合、インターフェイスで **vrf member** が変更されると、新しく作成された *vrf-name* も EIGRP プロセスのコンテキストで反映されます。

	コマンドまたはアクション	目的

	`version 10.3(1) Bios:version 05.47`
	`feature eigrp`
	
	`router eigrp 10`
	`  vrf eigrp-vrf`
	
	`interface Ethernet1/8`
	`  ip router eigrp 10`	

Step 8	（任意） **show {ip | ipv6} eigrp interfaces** 例： `switch(config-if)# show ip eigrp interfaces`	EIGRPインターフェイスに関する情報を表示します。
Step 9	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

例

（注） EIGRP プロセスを削除する場合は、インターフェイス モードで設定された EIGRP コマンドも削除する必要があります。

次に、EIGRP プロセスを作成し、EIGRP のインターフェイスを設定する例を示します。

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config-router)# interface ethernet 1/2
switch(config-if)# ip router eigrp Test1
switch(config-if)# no shutdown
switch(config-if)# copy running-config startup-config
```

その他の EIGRP パラメータの詳細については、高度な EIGRP の設定（255 ページ）の項を参照してください。

EIGRP インスタンスの再起動

EIGRP インスタンスを再起動できます。この処理では、インスタンスのすべてのネイバーが消去されます。

EIGRP インスタンスを再起動し、関連付けられたすべてのネイバーを削除するには、グローバル設定モードで次のコマンドを使用します。

手順の概要

1. （任意） **flush-routes**

2. **restart eigrp** *instance-tag*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	（任意） **flush-routes** 例： `switch(config)# flush-routes`	この EIGRP インスタンスを再起動するときに、ユニキャスト RIB のすべての EIGRP ルートをフラッシュします。
Step 2	**restart eigrp** *instance-tag* 例： `switch(config)# restart eigrp Test1`	EIGRPインスタンスを再起動して、すべてのネイバーを削除します。インスタンスタグには最大20文字の英数字を使用できます。大文字と小文字を区別します。

EIGRP インスタンスのシャットダウン

EIGRP インスタンスを正常にシャットダウンできます。これにより、すべてのルートと隣接関係は削除されますが、EIGRP 設定は保持されます。

EIGRPインスタンスをディセーブルにするには、ルータコンフィギュレーションモードで次のコマンドを使用します。

手順の概要

1. **shutdown**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**shutdown** 例： `switch(config-router)# shutdown`	この EIGRP インスタンスをディセーブルにします。EIGRP ルータ設定は残ります。

EIGRP のパッシブ インターフェイスの設定

EIGRP のパッシブ インターフェイスを設定できます。パッシブ インターフェイスは EIGRP 隣接関係に参加しませんが、このインターフェイスのネットワーク アドレスは EIGRP トポロジ テーブルに残ります。

EIGRPのパッシブインターフェイスを設定するには、インターフェイスコンフィギュレーションモードで次のコマンドを使用します。

手順の概要

1. {**ip** | **ipv6**} **passive-interface eigrp** *instance-tag*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	{**ip** | **ipv6**} **passive-interface eigrp** *instance-tag* 例： `switch(config-if)# ip passive-interface eigrp tag10`	EIGRP hello を抑制します。これにより、EIGRP インターフェイス上でネイバーがルーティングアップデートを形成および送信することを防ぎます。インスタンスタグには最大20文字の英数字を使用できます。大文字と小文字を区別します。

インターフェイスでの EIGRP のシャットダウン

インターフェイスでEIGRPを正常にシャットダウンできます。これにより、すべての隣接関係が削除され、このインターフェイスでEIGRPトラフィックが停止しますが、EIGRP設定は保持されます。

インターフェイスでEIGRPを無効にするには、インターフェイス設定モードで次のコマンドを使用します。

手順の概要

1. {**ip** | **ipv6**} **eigrp** *instance-tag* **shutdown**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	{**ip** | **ipv6**} **eigrp** *instance-tag* **shutdown** 例： `switch(config-if)# ip eigrp Test1 shutdown`	このインターフェイスで EIGRP を無効にします。EIGRP インターフェイス設定は残ります。インスタンスタグには最大20文字の英数字を使用できます。大文字と小文字を区別します。

高度な EIGRP の設定

EIGRP での認証の設定

EIGRP のネイバー間に認証を設定できます。「認証（Authentication）」セクションを参照してください。

EIGRP プロセスまたは個々のインターフェイスに対応する EIGRP 認証を設定できます。インターフェイスの EIGRP 認証設定は、EIGRP プロセスレベルの認証設定より優先されます。

始める前に

EIGRP 機能がイネーブルにする必要があります（「EIGRP 機能のいネーブル化（Enabling the EIGRP Feature）」を参照）。

EIGRP プロセスのすべてのネイバーが、共有認証キーを含め、同じ認証設定を共有することを確認します。

この認証設定のためのキー チェーンを作成します。詳細については、『Cisco Nexus 9000 Series NX-OS Security Configuration Guide』を参照してください。

手順の概要

1. **configure terminal**
2. **router eigrp** *instance-tag*
3. **address-family** {**ipv4** | **ipv6**} **unicast**
4. **authentication key-chain** *key-chain*
5. **authentication mode md5**
6. **interface** *interface-type slot/port*
7. {**ip** | **ipv6**} **router eigrp** *instance-tag*
8. {**ip** | **ipv6**} **authentication key-chain eigrp** *instance-tag keychain*
9. {**ip** | **ipv6**} **authentication mode eigrp** *instance-tag* **md5**
10. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的

Step 2	**router eigrp** *instance-tag*
	例：
	`switch(config)# router eigrp Test1`
	`switch(config-router)#`
		インスタンス タグを設定して、新しい EIGRP プロセスを作成します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。
		AS 番号であると認められていない *instance-tag* を設定する場合は、**autonomous-system** コマンドを使用して AS 番号を明示的に設定する必要があります。そうしないと、この EIGRP インスタンスはシャットダウン状態のままになります。

Step 3	**address-family** {**ipv4** | **ipv6**} **unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		アドレス ファミリ コンフィギュレーション モードを開始します。IPv4 の場合、このコマンドはオプションです。

Step 4	**authentication key-chain** *key-chain* 例： `switch(config-router-af)# authentication key-chain routeKeys`	この VRF の EIGRP プロセスにキーチェーンを関連付けます。キーチェーン名は、大文字と小文字が区別される 63 文字以下の任意の英数字文字列にできます。
Step 5	**authentication mode md5** 例： `switch(config-router-af)# authentication mode md5`	この VRF の MD5 メッセージ ダイジェスト認証モードを設定します。

Step 6	**interface** *interface-type slot/port*
	例：
	`switch(config-router-af) interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。**?** を使用すると、サポートされているインターフェイスを調べることができます。

Step 7	{**ip** | **ipv6**} **router eigrp** *instance-tag* 例： `switch(config-if)# ip router eigrp Test1`	このインターフェイスを、設定された EIGRP プロセスに関連付けます。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。
Step 8	{**ip** | **ipv6**} **authentication key-chain eigrp** *instance-tag keychain* 例： `switch(config-if)# ip authentication key-chain eigrp Test1 routeKeys`	このインターフェイスの EIGRP プロセスにキーチェーンを関連付けます。この設定は、ルータの VRF モードで設定された認証設定よりも優先します。 インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。

	コマンドまたはアクション	目的
Step 9	{**ip** | **ipv6**} **authentication mode eigrp** *instance-tag* **md5** 例： `switch(config-if)# ip authentication mode eigrp Test1 md5`	このインターフェイスの MD5 メッセージ ダイジェスト認証モードを設定します。この設定は、ルータの VRF モードで設定された認証設定よりも優先します。 インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。
Step 10	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

例

次に、EIGRP の MD5 メッセージ ダイジェスト認証をイーサネット インターフェイス 1/2 上で設定する例を示します。

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config-router)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# ip router eigrp Test1
switch(config-if)# ip authentication key-chain eigrp Test1 routeKeys
switch(config-if)# ip authentication mode eigrp Test1 md5
switch(config-if)# copy running-config startup-config
```

EIGRP スタブ ルーティングの設定

EIGRP スタブ ルーティング用のルータを設定できます。

ルータで EIGRP スタブ ルーティングを設定するには、アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **stub** [**direct** | **receive-only** | **redistributed** [**direct**] **leak-map** *map-name*]
2. （任意） **show ip eigrp neighbor detail**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**stub** [**direct** | **receive-only** | **redistributed** [**direct**] **leak-map** *map-name*] 例： `switch(config-router-af)# eigrp stub redistributed`	リモート ルータを EIGRP スタブ ルータとして設定します。リークマップのマップネーム名は、構成されたルートマップを参照します。複数のオプションを一度に構成して、目的のスタブ ルータ機能を有効にすることができます。
Step 2	（任意） **show ip eigrp neighbor detail** 例： `switch(config-router-af)# show ip eigrp neighbor detail`	ルータがスタブ ルータとして設定されていることを確認します。

例

次に、直接接続され、再配布されるルートをアドバタイズするスタブ ルータを設定する例を示します。

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config-router)# address-family ipv6 unicast
switch(config-router-af)# stub direct redistributed
switch(config-router-af)# copy running-config startup-config
```

ルータがスタブ ルータとして設定されていることを確認するには、**show ip eigrp neighbor detail** コマンドを使用します。出力の最後の行は、リモート ルータまたはスポーク ルータのスタブ ステータスを示します。

次に、**show ip eigrp neighbor detail** コマンドの出力例を示します。

```
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
```

EIGRP のサマリー アドレスの設定

指定したインターフェイスにサマリー集約アドレスを設定できます。より具体的なルートがルーティング テーブルにある場合、EIGRP は、より具体的なすべてのルートの最小に等しいメトリックを持つインターフェイスからのサマリー アドレスをアドバタイズします。「ルート集約」の項を参照してください。

サマリー集約アドレスを設定するには、インターフェイスコンフィギュレーションモードで次のコマンドを使用します。

手順の概要

1. {**ip** | **ipv6**} **summary-address eigrp** *instance-tag ip-prefix/length* [*distance* | **leak-map** *map-name*]

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	{**ip** | **ipv6**} **summary-address eigrp** *instance-tag ip-prefix/length* [*distance* | **leak-map** *map-name*] 例： `switch(config-if)# ip summary-address eigrp Test1 192.0.2.0/8`	サマリー集約アドレスを、IP プレフィックス/長さとして設定します。インスタンス タグおよびマップ名には、構成されたルートマップを参照する、最大 20 文字の英数字を使用できます。大文字と小文字は区別されます。 また、この集約アドレスのアドミニストレーティブディスタンスを設定することもできます。集約アドレスのデフォルトアドミニストレーティブディスタンスは 5 です。 （注） EIGRP がすでに実行されている場合を除き、プレフィックス/長さ形式をアドレス マスクの代わりに使用して IP アドレスを設定することを推奨します。EIGRP インスタンスが起動する前にアドレス マスク形式を使用すると、後でサマリーアドレスを削除または変更できなくなります。

例

この例は、EIGRP がネットワーク 192.0.2.0 をイーサネット 1/2 だけに集約するようにする方法を示しています。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if) ip summary-address eigrp Test1 192.0.2.0/24
```

EIGRP へのルートの再配布

他のルーティング プロトコルから EIGRP にルートを再配布できます。

始める前に

EIGRP 機能がイネーブルにする必要があります（「EIGRP 機能のいネーブル化（Enabling the EIGRP Feature）」を参照）。

他のプロトコルから再配布されるルートには、メトリック（デフォルト メトリック設定オプションまたはルート マップによる）を設定する必要があります。

ルート マップを作成して、EIGRP に再配布されるルートのタイプを管理する必要があります。Route Policy Manager の設定 （587 ページ）を参照してください。

手順の概要

1. **configure terminal**
2. **router eigrp** *instance-tag*
3. **address-family** {**ipv4** | **ipv6**} **unicast**
4. **redistribute** {**bgp** *as* | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **direct** | **static**} **route-map** *map-name*
5. **default-metric** *bandwidth delay reliability loading mtu*
6. （任意） **show** {**ip** | **ipv6**} **eigrp route-map statistics redistribute**
7. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router eigrp** *instance-tag*
	例：
	`switch(config)# router eigrp Test1`
	`switch(config-router)#`
		インスタンス タグを設定して、新しい EIGRP プロセスを作成します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。
		AS 番号であると認められていない *instance-tag* を設定する場合は、**autonomous-system** コマンドを使用して AS 番号を明示的に設定する必要があります。そうしないと、この EIGRP インスタンスはシャットダウン状態のままになります。

Step 3	**address-family** {**ipv4**	**ipv6**} **unicast**
		例：
		`switch(config-router)# address-family ipv4 unicast`
		`switch(config-router-af)#`
			アドレス ファミリ コンフィギュレーション モードを開始します。IPv4 の場合、このコマンドはオプションです。

	コマンドまたはアクション	目的
Step 4	**redistribute** {**bgp** *as* | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **direct** | **static**} **route-map** *map-name* 例： `switch(config-router-af)# redistribute bgp 100 route-map BGPFilter`	1 つのルーティング ドメインから EIGRP にルートを注入します。インスタンス タグおよびマップ名には最大 20 文字の英数字を使用できます。大文字と小文字は区別されます。
Step 5	**default-metric** *bandwidth delay reliability loading mtu* 例： `switch(config-router-af)# default-metric 500000 30 200 1 1500`	ルート再配布で学習したルートに割り当てられるメトリックを設定します。デフォルト値は次のとおりです。 • bandwidth：100000 Kbps • delay：100（10 マイクロ秒単位） • reliability：255 • loading：1 • MTU：1492
Step 6	（任意） **show** {**ip** | **ipv6**} **eigrp route-map statistics redistribute** 例： `switch(config-router-af)# show ip eigrp route-map statistics redistribute bgp`	EIGRP ルート マップ統計に関する情報を表示します。
Step 7	（任意） **copy running-config startup-config** 例： `switch(config-router-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、BGP を IPv4 向けの EIGRP に再配布する例を示します。

```
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
```

再配布されるルート数の制限

ルートの再配布では、多くのルートをEIGRPルートテーブルに追加できます。外部プロトコルから受け取るルートの数の上限を設定できます。EIGRP では、再配布されるルートの上限を設定するために次のオプションが用意されています。

- 固定制限：EIGRP は、構成された最大値まで再配布されたルートを受け入れます。デフォルトでは、EIGRP はデフォルトのしきい値である 75% を超えた場合、および最大制限に達した場合に警告メッセージをログに記録します。必要に応じて、最大再配布ルートのしきい値パーセンテージを構成できます。
- 警告のみ：設定された最大値のしきい値パーセンテージを超えた場合に、警告メッセージをログに記録します。ただし、EIGRP は再配布されたルートを受け入れ続けます。
- 取り消し：EIGRP が最大値に達したときにタイムアウト期間を開始します。タイムアウト期間の経過後、再配布されたルートの現在数が最大数よりも少ない場合、EIGRP はすべての再配布されたルートを要求します。再配布されたルートの現在数が最大数に達した場合、EIGRP はすべての再配布されたルートを取り消します。EIGRP が再配布されたルートをさらに受け入れられるように、この条件をクリアする必要があります。任意で、タイムアウト期間を設定できます。
- 最大プレフィックス値を、予想される再配布ルートの 2 倍に設定することを推奨します。
- ルート再配布は、8 個を超える redistribute コマンドをサポートしません。8 つのコマンドを構成した後、新しいルートはルーティングテーブルまたは動的ルーティングデータベースに追加されません。

（注） このタスクを設定できるのは、IPv4 VRF アドレス ファミリ コンフィギュレーション モードだけです。

始める前に

EIGRP 機能がイネーブルにする必要があります（「EIGRP 機能のいネーブル化（Enabling the EIGRP Feature）」を参照）。

手順の概要

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

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**router eigrp** *instance-tag*
	例：
	`switch(config)# router eigrp Test1`
	`switch(config-router)#`
		インスタンスタグを設定して、新しいEIGRPインスタンスを作成します。

Step 3	**redistribute** {**bgp** *id* | **direct** | **eigrp** *id* | **isis** *id* | **ospf** *id* | **rip** *id* | **static**} **route-map** *map-name* 例： `switch(config-router)# redistribute bgp route-map FilterExternalBGP`	設定したルート マップ経由で、選択したプロトコルを EIGRP に再配布します。
Step 4	**redistribute maximum-prefix** *max* [*threshold*] [**warning-only** | **withdraw** [*num-retries timeout*]] 例： `switch(config-router)# redistribute maximum-prefix 1000 75 warning-only`	EIGRP が配布するプレフィックスの最大数を指定します。有効な範囲は 1 ～ 65535 です。任意で次のオプションを指定します。 • *threshold*：警告メッセージをトリガーする最大プレフィックス数のパーセンテージ。 • **warning-only**：プレフィックスの最大数を超えた場合に警告メッセージを記録します。 • **withdraw**：再配布されたすべてのルートを取り消します。任意で再配布されたルートを取得しようと試みます。*num-retries* の範囲は 1 ～ 12。*timeout* は 60 ～ 600 秒です。デフォルトは 300 秒です。 **clear ip eigrp redistribution** コマンドを使用し、 すると、すべてのルートを取り消すことができます。 （注） EIGRP トポロジでは、最大プレフィックス値を予想される再配布ルートの 2 倍に設定することをお勧めします。
Step 5	（任意） **show running-config eigrp** 例：	EIGRP の設定を表示します。

	コマンドまたはアクション	目的
	`switch(config-router)# show running-config eigrp`	
Step 6	（任意） **copy running-config startup-config** 例： `switch(config-router)# copy running-config startup-config`	この設定変更を保存します。

例

次に、EIGRP に再配布されるルートの数を制限する例を示します。

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config-router)# redistribute bgp route-map FilterExternalBGP
switch(config-router)# redistribute maximum-prefix 1000 75
```

EIGRP でのロードバランスの設定

EIGRP でのロードバランスを設定できます。**maximum-paths** オプションを使用して、等コストマルチパス（ECMP）のルート数を設定できます。「EIGRP でのロードバランスの設定」の項を参照してください。

始める前に

EIGRP 機能が有効にする必要があります（「EIGRP 機能のいネーブル化（Enabling the EIGRP Feature）」を参照）。

手順の概要

1. **configure terminal**
2. **router eigrp** *instance-tag*
3. **address-family** {**ipv4** | **ipv6**} **unicast**
4. **maximum-paths** *num-paths*
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

	コマンドまたはアクション	目的

Step 2	**router eigrp** *instance-tag*
	例：
	`switch(config)# router eigrp Test1`
	`switch(config-router)#`
		インスタンスタグを設定して、新しいEIGRPプロセスを作成します。インスタンスタグには最大20文字の英数字を使用できます。大文字と小文字を区別します。
		AS 番号であると認められていない *instance-tag* を設定する場合は、**autonomous-system** コマンドを使用して AS 番号を明示的に設定する必要があります。そうしないと、この EIGRP インスタンスはシャットダウン状態のままになります。

Step 3	**address-family** {**ipv4** | **ipv6**} **unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		アドレスファミリ コンフィギュレーションモードを開始します。IPv4 の場合、このコマンドはオプションです。

Step 4	**maximum-paths** *num-paths* 例： `switch(config-router-af)# maximum-paths 5`	EIGRPがルートテーブルに受け入れる等コストパスの数を設定します。指定できる範囲は 1 ～ 32 です。デフォルト値は 8 です。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config-router-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、6 つまでの等コストパスによる、EIGRP の等コスト ロードバランスを IPv4 上で設定する例を示します。

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config-router)# maximum-paths 6
switch(config-router)# copy running-config startup-config
```

EIGRP のグレースフル リスタートの設定

EIGRPに対してグレースフルリスタートまたはノンストップフォワーディングを設定できます。「グレースフル リスタートおよびハイ アベイラビリティ」を参照してください。

（注） デフォルトでは、グレースフル リスタートはイネーブルです。

始める前に

EIGRP 機能がイネーブルにする必要があります（「EIGRP 機能のいネーブル化（Enabling the EIGRP Feature）」を参照）。

NSF 認識ルータが動作中であり、ネットワークで完全に収束している場合にのみ、このルータが NSF 対応ルータのグレースフル リスタート動作を支援できます。

グレースフルリスタートに参加するネイバーデバイスは、NSF認識またはNSF対応である必要があります。

手順の概要

1. **configure terminal**
2. **router eigrp** *instance-tag*
3. **address-family** {**ipv4** | **ipv6**} **unicast**
4. **graceful-restart**
5. **timers nsf converge** *seconds*
6. **timers nsf route-hold** *seconds*
7. **timers nsf signal** *seconds*
8. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例： `switch# configure terminal`	グローバル コンフィギュレーション モードを開始します。

Step 2	**router eigrp** *instance-tag*
	例：
	`switch(config)# router eigrp Test1`
	`switch(config-router)#`
		インスタンス タグを設定して、新しい EIGRP プロセスを作成します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。
		AS 番号であると認められていない *instance-tag* を設定する場合は、**autonomous-system** コマンドを使用して AS 番号を明示的に設定する必要があります。そうしないと、この EIGRP インスタンスはシャットダウン状態のままになります。

Step 3	**address-family** {**ipv4** | **ipv6**} **unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		アドレス ファミリ コンフィギュレーション モードを開始します。IPv4 の場合、このコマンドはオプションです。

	コマンドまたはアクション	目的
Step 4	**graceful-restart** 例： `switch(config-router-af)# graceful-restart`	グレースフル リスタートをイネーブルにします。この機能は、デフォルトでイネーブルにされています。
Step 5	**timers nsf converge** *seconds* 例： `switch(config-router-af)# timers nsf converge 100`	スイッチオーバー後にコンバージェンスするまでの制限時間を設定します。範囲は 60 ～ 180 秒です。デフォルトは 120 です。
Step 6	**timers nsf route-hold** *seconds* 例： `switch(config-router-af)# timers nsf route-hold 200`	グレースフル リスタート認識ピアから学習したルートのホールド タイムを設定します。範囲は 20 ～ 300 秒です。デフォルトは 240 です。
Step 7	**timers nsf signal** *seconds* 例： `switch(config-router-af)# timers nsf signal 15`	グレースフル リスタートの信号を送信する時間制限を設定します。範囲は 10 ～ 30 秒です。デフォルトは 20 です。
Step 8	（任意） **copy running-config startup-config** 例： `switch(config-router-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、デフォルト タイマー値を使用して IPv6 上で EIGRP のグレースフル リスタートを設定する例を示します。

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config-router)# address-family ipv6 unicast
switch(config-router-af)# graceful-restart
switch(config-router-af)# copy running-config startup-config
```

hello パケット間のインターバルとホールド タイムの調整

Hello メッセージの間隔とホールド タイムを調整できます。

デフォルトでは、5 秒ごとに Hello メッセージが送信されます。ホールド タイムは Hello メッセージでアドバタイズされ、送信者が有効であると見なすまでの時間をネイバーに示します。デフォルトの保留時間は、hello 間隔の 3 倍（15 秒）です。

非常に輻輳した大規模なネットワークでは、デフォルトの保留時間では、全ルータがネイバーから hello パケットを受信するまでに十分な時間がない場合もあります。この場合は、ホールド タ

イムを増やすことを推奨します。ホールド タイムを変更するには、インターフェイス コンフィギュレーション モードでステップ 2 のコマンドを使用します。

手順の概要

1. {**ip** | **ipv6**} **hello-interval eigrp** *instance-tag seconds*
2. {**ip** | **ipv6**} **hold-time eigrp** *instance-tag seconds*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	{**ip** | **ipv6**} **hello-interval eigrp** *instance-tag seconds* 例： `switch(config-if)# ip hello-interval eigrp Test1 30`	EIGRP ルーティング処理の hello 間隔を設定します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。範囲は 1 ～ 65535 秒です。デフォルトは 5 分です。
Step 2	{**ip** | **ipv6**} **hold-time eigrp** *instance-tag seconds* 例： `switch(config-if)# ipv6 hold-time eigrp Test1 30`	EIGRP ルーティング処理のホールド タイムを設定します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。範囲は 1 ～ 65535 秒です。

例

タイマー設定を確認するには、**show ip eigrp interface detail** コマンドを使用します。

スプリット ホライズンの無効化

スプリット ホライズンを使用すると、ルータによって情報元インターフェイスからルート情報がアドバタイズされないようにできます。通常はスプリット ホライズンにより、特にリンクに障害がある場合に、複数のルーティング デバイス間での通信が最適化されます。

デフォルトでは、スプリット ホライズンはすべてのインターフェイスで有効になっています。

スプリット ホライズンを無効にするには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **no** {**ip** | **ipv6**} **split-horizon eigrp** *instance-tag*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**no** {**ip** | **ipv6**} **split-horizon eigrp** *instance-tag* 例： `switch(config-if)# no ip split horizon eigrp Test1`	スプリット ホライズンを無効にします。

ワイド メトリックスの有効化

ワイド メトリックを有効化し、オプションとして RIB のスケール係数を設定するには、ルータ設定モードまたはアドレス ファミリ設定モードで次のコマンドを使用します。

手順の概要

1. **metrics version 64bit**
2. （任意） **metrics rib-scale** *value*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**metrics version 64bit** 例： `switch(config-router)# metrics version 64bit`	64 ビット メトリック値を有効にします。
Step 2	（任意） **metrics rib-scale** *value* 例： `switch(config-router)#`	RIB で 64 ビットのメトリック値を 32 ビットに変換するために使用されるスケール係数を設定します。範囲は 1 ～ 255 です。デフォルト値は 128 です。

EIGRP の調整

オプション パラメータを設定し、ネットワークに合わせて EIGRP を調整できます。

アドレス ファミリ コンフィギュレーション モードでは、次のオプション パラメータを設定できます。

手順の概要

1. **default-information originate** [**always** | **route-map** *map-name*]
2. **distance** *internal external*

3. **metric max-hops** *hop-count*
4. **metric weights** *tos k1 k2 k3 k4 k5 k6*
5. **nsf await-redist-proto-convergence**
6. **timers active-time** {*time-limit* | **disabled**}
7. （任意）{**ip** | **ipv6**} **bandwidth eigrp** *instance-tag bandwidth*
8. {**ip** | **ipv6**} **bandwidth-percent eigrp** *instance-tag percent*
9. [**no**] {**ip** | **ipv6**} **delay eigrp** *instance-tag delay*
10. {**ip** | **ipv6**} **distribute-list eigrp** *instance-tag* {**prefix-list** *name* | **route-map** *map-name*} {**in** | **out**}
11. [**no**] {**ip** | **ipv6**} **next-hop-self eigrp** *instance-tag*
12. {**ip** | **ipv6**} **offset-list eigrp** *instance-tag* {**prefix-list** *name* | **route-map** *map-name*} {**in** | **out**} *offset*
13. {**ip** | **ipv6**} **passive-interface eigrp** *instance-tag*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**default-information originate** [**always** | **route-map** *map-name*] 例： `switch(config-router-af)# default-information originate always`	プレフィックス 0.0.0.0/0 を持つデフォルトルートを発信するか、受け入れます。ルートマップが提供されると、ルートマップが true 状態となっている場合にのみデフォルト ルートが発信されます。ルートマップ名には最大 20 文字の英数字を使用できます。大文字と小文字は区別されます。
Step 2	**distance** *internal external* 例： `switch(config-router-af)# distance 25 100`	この EIGRP プロセスのアドミニストレーティブディスタンスを設定します。範囲は 1 ～ 255 です。内部の値で、同じ自律システム内で学習したルートのディスタンスが設定されます（デフォルト値は 90 です）。外部の値で、外部自律システムから学習したルートのディスタンスが設定されます（デフォルト値は 170 です）。
Step 3	**metric max-hops** *hop-count* 例： `switch(config-router-af)# metric max-hops 70`	アドバタイズされるルートに許容される最大ホップ数を設定します。ホップ数がこの最大値を超えるルートは、到達不能としてアドバタイズされます。範囲は 1 ～ 255 です。デフォルトは 100 です。
Step 4	**metric weights** *tos k1 k2 k3 k4 k5 k6* 例： `switch(config-router-af)# metric weights 0 1 3 2 1 0`	EIGRP メトリックまたは K 値を調整します。EIGRP は次の式を使用して、ネットワークへの合計メトリックを決定します。 メトリック = [k1 × 帯域幅 +（k2 × 帯域幅）/（256 – 負荷）+ k3 × 遅延 + k6 × 拡張属性] x [k5/（信頼性 + k4）]

	コマンドまたはアクション	目的
		デフォルト値と指定できる範囲は、次のとおりです。 • TOS：0。指定できる範囲は 0 ～ 8 です。 • k1：1。有効な範囲は 0 ～ 255 です。 • k2：0。有効な範囲は 0 ～ 255 です。 • k3：1。有効な範囲は 0 ～ 255 です。 • k4：0。有効な範囲は 0 ～ 255 です。 • k5：0。有効な範囲は 0 ～ 255 です。 • k6：0。有効な範囲は 0 ～ 255 です。
Step 5	**nsf await-redist-proto-convergence** 例： `switch(config-router-af)# nsf await-redist-proto-convergence`	ノンストップフォワーディング（NSF）中に、EIGRPがルーティング情報ベース（RIB）に独自のルートをインストールする前に、再配布されたプロトコルのコンバージェンスを待機します。 このコマンドは、NSFが進行中で、BGPが収束してルートをインストールするまでEIGRPが待機するスイッチオーバーシナリオで役立ちます。これにより、BGPが収束し、EIGRPが宛先への代替パスを見つける前に、EIGRPが一時的なルートをインストールして転送情報ベース（FIB）エントリを変更することを防止できます。 （注） EIGRPとBGPの間で相互再配布が設定されている場合（PE-CE環境など）にこのコマンドを使用すると、プロバイダーエッジ（PE）ルータがBGPまでRIBにEIGRPルートをインストールしないため、トラフィック損失が発生する可能性があります。ルートを使用できます。この動作により、カスタマーエッジ（CE）ルータがEIGRPから学習し、ピアPEルータにアドバタイズするルートが遅延します。
Step 6	**timers active-time** {*time-limit* | **disabled**} 例： `switch(config-router-af)# timers active-time 200`	（照会の送信後に）ルートがアクティブ（SIA）状態のままとなっていることを宣言するまでに、ルータが待機する時間を分単位で設定します。有効な範囲は 1 ～ 65535 です。デフォルトは 3 です。
Step 7	（任意）{**ip** | **ipv6**} **bandwidth eigrp** *instance-tag bandwidth* 例：	インターフェイス上の EIGRP の帯域幅メトリックを設定します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別し

	コマンドまたはアクション	目的
	`switch(config-if)# ip bandwidth eigrp Test1 30000`	ます。帯域幅の範囲は、1 ～ 2,560,000,000 kbps です。
Step 8	{**ip** | **ipv6**} **bandwidth-percent eigrp** *instance-tag percent* 例： `switch(config-if)# ip bandwidth-percent eigrp Test1 30`	EIGRP がインターフェイス上で使用する可能性のある帯域幅の割合を設定します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。割合の範囲は 0 ～ 100 です。デフォルトは 50 です。
Step 9	[**no**] {**ip** | **ipv6**} **delay eigrp** *instance-tag delay* 例： `switch(config-if)# ip delay eigrp Test1 100`	インターフェイス上の EIGRP の遅延メトリックを設定します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。遅延の範囲は、1 ～ 16777215（10 マイクロ秒単位）です。
Step 10	{**ip** | **ipv6**} **distribute-list eigrp** *instance-tag* {**prefix-list** *name* | **route-map** *map-name*} {**in** | **out**} 例： `switch(config-if)# ip distribute-list eigrp Test1 route-map EigrpTest in`	このインターフェイス上の EIGRP のルータ フィルタリング ポリシーを設定します。インスタンス タグ、プレフィックス リスト名、およびルート -マップ名には最大 20 文字の英数字を使用できます。大文字と小文字は区別されます。
Step 11	[**no**] {**ip** | **ipv6**} **next-hop-self eigrp** *instance-tag* 例： `switch(config-if)# ipv6 next-hop-self eigrp Test1`	このインターフェイスのアドレスではなく、受信したネクストホップアドレスを使用するよう、EIGRP を設定します。デフォルトでは、このインターフェイスの IP アドレスをネクストホップ アドレスに使用します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。
Step 12	{**ip** | **ipv6**} **offset-list eigrp** *instance-tag* {**prefix-list** *name* | **route-map** *map-name*} {**in** | **out**} *offset* 例： `switch(config-if)# ip offset-list eigrp Test1 prefix-list EigrpList in`	EIGRP が学習したルートに、着信および発信メトリックへのオフセットを追加します。インスタンス タグ、プレフィックス リスト名、およびルート-マップ名には最大 20 文字の英数字を使用できます。大文字と小文字は区別されます。
Step 13	{**ip** | **ipv6**} **passive-interface eigrp** *instance-tag* 例： `switch(config-if)# ip passive-interface eigrp Test1`	EIGRP hello を抑制します。これにより、EIGRP インターフェイス上でネイバーがルーティング アップデートを形成および送信することを防ぎます。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。

EIGRP の仮想化の設定

複数の VRF を作成して、各 VRF で同じまたは複数の EIGRP プロセスを使用することもできます。VRF にはインターフェイスを割り当てます。

（注） インターフェイスの VRF を設定した後に、インターフェイスの他のすべてのパラメータを設定します。インターフェイスの VRF を設定すると、そのインターフェイスの他の設定がすべて削除されます。

始める前に

EIGRP 機能が有効にする必要があります（「EIGRP 機能のいネーブル化（Enabling the EIGRP Feature）」を参照）。

VRF を作成します。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **router eigrp** *instance-tag*
4. **interface ethernet** *slot//port*
5. **vrf member** *vrf-name*
6. {**ip** | **ipv6**} **router eigrp** *instance-tag*
7. **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`	

Step 2	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context RemoteOfficeVRF`
	`switch(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。VRF 名には最大 20 文字の英数字を使用できます。大文字と小文字は区別されます。

	コマンドまたはアクション	目的

Step 3	**router eigrp** *instance-tag*
	例：
	`switch(config-vrf)# router eigrp Test1`
	`switch(config-router)#`
		インスタンスタグを設定して、新しいEIGRPプロセスを作成します。インスタンスタグには最大20文字の英数字を使用できます。大文字と小文字を区別します。
		AS 番号であると認められていない *instance-tag* を設定する場合は、**autonomous-system** コマンドを使用して AS 番号を明示的に設定する必要があります。そうしないと、このEIGRPインスタンスはシャットダウン状態のままになります。

Step 4	**interface ethernet** *slot//port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイスコンフィギュレーションモードを開始します。**?** を使用すると、スロットおよびポートの範囲を検索できます。

Step 5	**vrf member** *vrf-name* 例： `switch(config-if)# vrf member RemoteOfficeVRF`	このインターフェイスをVRFに追加します。VRF名には最大20文字の英数字を使用できます。大文字と小文字は区別されます。
Step 6	{**ip** | **ipv6**} **router eigrp** *instance-tag* 例： `switch(config-if)# ip router eigrp Test1`	このインターフェイスをEIGRPプロセスに追加します。インスタンスタグには最大20文字の英数字を使用できます。大文字と小文字を区別します。
Step 7	**copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

例

次に、VRF を作成して、その VRF にインターフェイスを追加する例を示します。

```
switch# configure terminal
switch(config)# vrf context NewVRF
switch(config-vrf)# router eigrp Test1
switch(config-router)# interface ethernet 1/2
switch(config-if)# ip router eigrp Test1
switch(config-if)# vrf member NewVRF
switch(config-if)# copy running-config startup-config
```

EIGRP の設定の確認

EIGRP の設定情報を表示するには、次のいずれかの作業を行います。

コマンド	目的
show {**ip** | **ipv6**} **eigrp** [*instance-tag*]	設定したEIGRPプロセスの要約を表示します。
show {**ip** | **ipv6**} **eigrp** [*instance-tag*] **interfaces** [*type number*] [**brief**] [**detail**]	設定されているすべてのEIGRPインターフェイスに関する情報を表示します。
show {**ip** | **ipv6**} **eigrp** *instance-tag* **neighbors** [*type number*] [**detail**]	すべてのEIGRPネイバーに関する情報を表示します。EIGRPネイバーの設定を確認するには、このコマンドを使用します。
show {**ip** | **ipv6**} **eigrp** [*instance-tag*] **route** [*ip-prefix/length*] [**active**] [**all-links**] [**detail-links**] [**pending**] [**summary**] [**zero-successors**] [**vrf** *vrf-name*]	すべてのEIGRPルートに関する情報を表示します。
show {**ip** | **ipv6**} **eigrp** [*instance-tag*] **topology** [*ip-prefix/length*] [**active**] [**all-links**] [**detail-links**] [**pending**] [**summary**] [**zero-successors**] [**vrf** *vrf-name*]	EIGRPトポロジテーブルに関する情報を表示します。
show running-configuration eigrp	現在実行中のEIGRPコンフィギュレーションを表示します。

EIGRP のモニタリング

EIGRP 統計情報を表示するには、次のコマンドを使用します。

コマンド	目的
show {**ip** | **ipv6**} **eigrp** [*instance-tag*] **accounting** [**vrf** *vrf-name*]	EIGRP の課金統計情報を表示します。
show {**ip** | **ipv6**} **eigrp** [*instance-tag*] **route-map statistics redistribute**	EIGRP の再配布統計情報を表示します。
show {**ip** | **ipv6**} **eigrp** [*instance-tag*] **traffic** [**vrf** *vrf-name*]	EIGRPのトラフィック統計情報を表示します。

EIGRP の設定例

次に、EIGRP を設定する例を示します。

```
feature eigrp
interface ethernet 1/2
 ip address 192.0.2.55/24
 ip router eigrp Test1
  no shutdown
router eigrp Test1
  router-id 192.0.2.1
```

次に、**distribute-list** でルートマップを使用する例を示します。 EIGRP ピアから動的に受信（またはアドバタイズ）されたルートをフィルタリングするコマンド。例では、EIGRP の外部プロトコルメトリックルートを、有効な偏差の 100、BGP のソースプロトコル、および自律システム 45000 と照合するための、ルートマップの設定をします。2 つの match 句が true の場合、対象のルーティングプロトコルのタグ値が 5 に設定されます。ルートマップを使用して、着信パケットを EIGRP プロセスへ配布します。

```
switch(config)# route-map metric-range
switch(config-route-map)# match metric external 500 +- 100

switch(config-route-map)# set tag 5
switch(config-route-map)# exit
switch(config)# router eigrp 1
switch(config-router)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# ip address 172.16.0.0
switch(config-if)# ip router eigrp 1
switch(config-if)# ip distribute-list eigrp 1 route-map metric-range in
```

次の例は、EIGRP トポロジテーブルに許可される前に、ルートマップでフィルタリングされるルーティングテーブルから再配布されるルートが受け入れられるよう、redistribute コマンドでルートマップを使用する方法を示します。この例は、EIGRP ルートを、110、200、または 700 ～ 800 の範囲のメトリックと照合するために、ルートマップを設定する方法を示しています。この match 句が true の場合、対象のルーティングプロトコルのタグ値が 10 に設定されます。ルートマップを使用して、EIGRP パケットを再配布します。

```
switch(config)# route-map metric-eigrp
switch(config-route-map)# match metric 110 200 750 +- 50
switch(config-route-map)# set tag 10
switch(config-route-map)# exit
switch(config)# router eigrp 1
switch(config-router)# redistribute eigrp route-map metric-eigrp
switch(config-router)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# ip address 172.16.0.0
switch(config-if)# ip router eigrp 1
```

関連項目

ルートマップの詳細については、Route Policy Manager の設定（587 ページ）を参照してください。

その他の参考資料

EIGRP の実装に関する詳細情報については、次のページを参照してください。

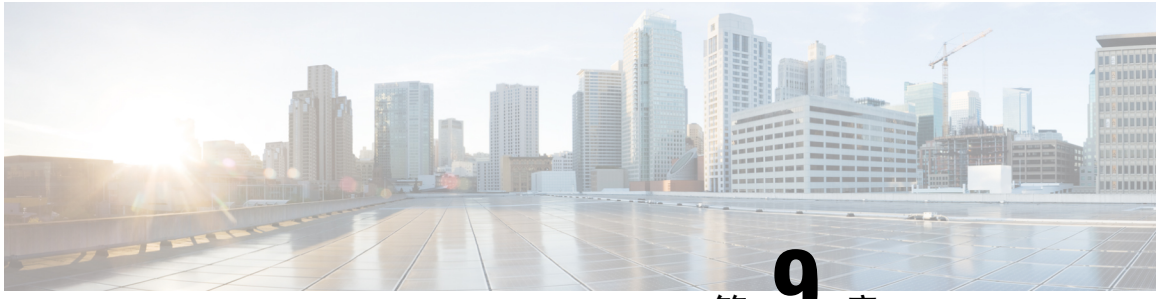
関連資料

関連項目	マニュアル タイトル
EIGRP CLI コマンド	*Cisco Nexus 9000* シリーズ *NX-OS* ユニキャスト ルーティング コマンド リファレンス
EIGRP テクニカル ノートの概要	EIGRP テクニカル ノートの概要
EIGRP よく寄せられる質問（FAQ）	EIGRP よく寄せられる質問（FAQ）

MIB

MIB	MIB のリンク
EIGRP に関連する MIB	MIB を検索およびダウンロードするには、次の URL にアクセスしてください。http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

MIB



第 9 章

IS-IS の設定

この章では、Cisco NX-OS デバイスの Integrated Intermediate System-to-Intermediate System（IS-IS）を設定する方法について説明します。

この章は、次の項で構成されています。

- IS-IS について （279 ページ）
- IS-IS 認証 （282 ページ）
- メッシュ グループ （282 ページ）
- 過負荷ビット （283 ページ）
- ルート集約 （283 ページ）
- ルートの再配布 （283 ページ）
- プレフィックスの抑制のリンク （284 ページ）
- ロード バランシング （284 ページ）
- BFD （284 ページ）
- 仮想化のサポート （284 ページ）
- 高可用性およびグレースフル リスタート （285 ページ）
- 複数の IS-IS インスタンス （285 ページ）
- IS-IS の前提条件 （285 ページ）
- IS-IS に関する注意事項および制限事項 （286 ページ）
- デフォルト設定 （286 ページ）
- IS-IS の設定 （287 ページ）
- IS-IS 設定の確認 （315 ページ）
- IS-IS の監視 （316 ページ）
- IS-IS の設定例 （317 ページ）
- 関連項目 （317 ページ）

IS-IS について

IS-IS は、ISO（国際標準化機構）/IEC（国際電気標準化会議）10589 に基づく IGP です。Cisco NX-OS は、インターネット プロトコル バージョン 4（IPv4）および IPv6 をサポートします。IS-IS はネットワーク トポロジの変化を検出し、ネットワーク上の他のノードへのループフリー ルート

を計算できる、ダイナミック リンクステート ルーティング プロトコルです。各ルータは、ネットワークの状態を記述するリンクステート データベースを維持し、設定された各リンクにパケットを送信してネイバーを検出します。IS-IS はネットワークを介して各ネイバーにリンクステート情報をフラッディングします。ルータもすべての既存ネイバーを通じて、リンクステート データベースのアドバタイズメントおよびアップデートを送信します。

IS-IS の概要

IS-IS は、設定されている各インターフェイスに hello パケットを送信し、IS-IS ネイバー ルータを検出します。hello パケットには認証、エリア、サポート対象プロトコルなど、受信側インターフェイスが発信側インターフェイスとの互換性を判別するために使用する情報が含まれます。また、一致する最大転送ユニット（MTU）設定を持つインターフェイスだけを使用して IS-IS が隣接関係を確立できるように、hello パケットがパディングされます。互換インターフェイスは隣接関係を形成し、リンクステート アップデート メッセージ（LSP）を使用して、リンクステート データベースのルーティング情報をアップデートします。ルータはデフォルトで、10 分間隔で定期的に LSP リフレッシュを送信し、LSP は 20 分間（LSP ライフタイム）リンクステート データベースに残ります。LSP ライフタイムが終了するまでにルータが LSP リフレッシュを受信しなかった場合、ルータはデータベースから LSP を削除します。

LSP 間隔は、LSP ライフタイムより短くする必要があります。そうしないと、リフレッシュ前に LSP がタイムアウトします。

IS-IS は、隣接ルータに定期的に hello パケットを送信します。hello パケットに対して一時モードを設定すると、IS-IS が隣接関係を確立する前に使用された余分なパディングがこれらの hello パケットに含まれなくなります。隣接ルータの MTU 値が変更された場合、IS-IS はこの変更を検出し、パディングされた hello パケットを一定期間送信できます。IS-IS はこの機能を使用して、隣接ルータ上の一致しない MTU 値を検出します。詳細については、「Hello パディングの一時モードの設定」の項を参照してください。

IS-IS エリア

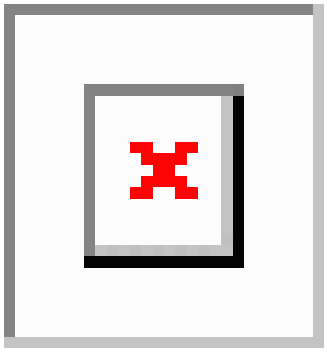
IS-IS ネットワークは、ネットワーク内のすべてのルータを含むシングルエリアとして設計することもできますし、バックボーンまたはレベル 2 エリアに接続する複数のエリアとして設計することもできます。非バックボーン エリアのルータはレベル 1 ルータで、ローカル エリア内で隣接関係を確立します（エリア内ルーティング）。レベル 2 エリアのルータは、他のレベル 2 ルータと隣接関係を確立し、レベル 1 エリア間のルーティングを実行します（エリア間ルーティング）。1 つのルータにレベル 1 エリアとレベル 2 エリアの両方を設定できます。これらのレベル 1/レベル 2 ルータは、エリア境界ルータとして動作し、ローカル エリアからレベル 2 バックボーン エリアに情報をルーティングします（下図を参照）。

レベル 1 エリア内のルータは、そのエリア内の他のすべてのルータに対する到達方法を認識します。レベル 2 ルータは、他のエリア境界ルータおよび他のレベル 2 ルータへの到達方法を認識します。レベル 1/レベル 2 ルータは 2 つのエリアの境界にまたがり、レベル 2 バックボーン エリアとの間で双方向にトラフィックをルーティングします。レベル 1/レベル 2 ルータはレベル 1 ルータの Attached（ATT）ビット信号を使用して、レベル 2 エリアに接続するため、このレベル 1/レベル 2 ルータへのデフォルト ルートを設定します。

エリア内に 2 台以上のレベル 1/レベル 2 ルータがある場合など、場合によっては、レベル 1 ルータがレベル 2 エリアへのデフォルト ルートとして使用するレベル 1/レベル 2 ルータを制御することもできます。Attached ビットを設定するレベル 1/レベル 2 ルータを設定できます。詳細については、「IS-IS 設定の確認」の項を参照してください。

Cisco NX-OS の IS-IS インスタンスは、レベル 1 またはレベル 2 エリアを 1 つだけサポートするか、またはそれぞれのエリアを 1 つずつサポートします。デフォルトでは、すべての IS-IS インスタンスが自動的にレベル 1 およびレベル 2 ルーティングをサポートします。

図 *26 :* エリアに分割された *IS-IS* ネットワーク



ASBR（自律システム境界ルータ）は、IS-IS AS（自律システム）全体に外部宛先をアドバタイズします。外部ルートは、他のプロトコルから IS-IS に再配布されたルートです。

NET およびシステム ID

IS-IS インスタンスごとにネットワーク エンティティ タイトル（NET）が関連付けられています。NET は、その IS-IS インスタンスをエリア内で一意に特定する IS-IS システム ID とエリア ID からなります。たとえば、NET が 47.0004.004d.0001.0001.0c11.1111.00 の場合、システム ID は 0000.0c11.1111.00、エリア ID は 47.0004.004d.0001 です。

DIS

IS-IS はブロードキャスト ネットワーク内で代表中継システム（DIS）を使用することにより、各ルータがブロードキャスト ネットワーク上の他のルータと不要なリンクを形成しないようにします。IS-IS ルータは DIS に LSP を送信し、DIS がブロードキャスト ネットワークのあらゆるリンクステート情報を管理します。エリア内で DIS を選択するために IS-IS に使用させる IS-IS プライオリティをユーザ側で設定できます。

（注） ポイントツーポイント ネットワークでは DIS は不要です。

IS-IS 認証

隣接関係および LSP 交換を制御するために、認証を設定できます。ネイバーになろうとするルータは、設定されている認証レベルの同じパスワードを交換する必要があります。パスワードが無効なルータは、IS-IS によってブロックされます。IS-IS 認証はグローバルに設定することも、レベル 1、レベル 2、またはレベル 1/レベル 2 両方のルーティングに対応する個々のインターフェイスに設定することもできます。

IS-IS がサポートする認証方式は、次のとおりです。

- クリア テキスト：交換するすべてのパケットで、クリアテキストの 128 ビット パスワードが伝送されます。
- MD5 ダイジェスト：交換するすべてのパケットで、128 ビット キーに基づくメッセージ ダイジェストが伝送されます。

受動的攻撃から保護するために、IS-IS はネットワークを介してクリア テキストとして MD5 秘密キーを送信します。また、リプレイ アタックから保護するために、IS-IS は各パケットにシーケンス番号を組み込みます。

hello および LSP 認証用のキーチェーンも使用できます。キーチェーン管理の詳細については、「Cisco Nexus 9000 シリーズ NX-OS セキュリティ設定ガイド」を参照してください。

メッシュ グループ

メッシュ グループは一連のインターフェイスであり、グループ内では、インターフェイスを介して到達可能なすべてのルータが他の各ルータとの間に 1 つ以上のリンクを持ちます。多数のリンクで障害が発生しても、ネットワークから 1 つまたは複数のルータが切り離されることはありません。

通常のフラッディングでは、新しい LSP を受信したインターフェイスは、その LSP をルータ上の他のすべてのインターフェイスにフラッディングします。メッシュ グループを使用する場合、メッシュ グループに含まれているインターフェイスは新しい LSP を受信しても、メッシュ グループ内の他のインターフェイスには、新しい LSP をフラッディングしません。

（注） 特定のメッシュ ネットワーク トポロジーで、ネットワークのスケーラビリティを向上させるために、LSP を制限しなければならない場合があります。LSP フラッディングを制限すると、ネットワークの信頼性も下がります（障害発生時）。したがって、メッシュ グループはどうしても必要な場合に限り、慎重にネットワークを設計したうえで使用することを推奨します。

ルータ間のパラレル リンクに、ブロック モードでメッシュ グループを設定することもできます。このモードでは、各ルータがそれぞれリンクステート情報を最初に交換すると、それ以後はメッシュ グループのそのインターフェイスですべての LSP がブロックされます。

過負荷ビット

IS-IS は過負荷ビットを使用して他のルータに指示を与え、それらのルータがトラフィックの転送にローカルルータを使用せずに、引き続きローカルルータ宛てのトラフィックをルーティングするようにします。

過負荷ビットを使用する状況は、次のとおりです。

- ルータがクリティカル条件下にある。
- ネットワークに対して通常手順でルータの追加および除去を行う。
- その他（管理上またはトラフィック エンジニアリング上）の理由。BGP コンバージェンスの待機中など。

ルート集約

サマリー集約アドレスを設定できます。ルート集約を使用すると、固有性の強い一連のアドレスをすべての固有アドレスを代表する 1 つのアドレスに置き換えることによって、ルート テーブルを簡素化できます。たとえば、10.1.1.0/24、10.1.2.0/24、および 10.1.3.0/24 というアドレスを 1 つの集約アドレス 10.1.0.0/16 に置き換えることができます。

IS-IS はルーティング テーブルに含まれている固有性の強いルートが多いほど、固有性の強いルートの最小メトリックと同じメトリックを指定して、サマリー アドレスをアドバタイズします。

（注） Cisco NX-OS は、自動ルート集約をサポートしていません。

ルートの再配布

IS-IS を使用すると、スタティック ルート、他の IS-IS 自律システムが学習したルート、または他のプロトコルからのルートを再配布できます。再配布を指定したルート マップを設定して、どのルートが IS-IS に渡されるかを制御する必要があります。ルート マップを使用すると、宛先、送信元プロトコル、ルートタイプ、ルートタグなどの属性に基づいて、ルートをフィルタリングできます。詳細については、Route Policy Manager の設定 （587 ページ）を参照してください。

IS-IS ルーティング ドメインにルートを再配布しても、デフォルトでは Cisco NX-OS がそのつど、IS-IS ルーティング ドメインにデフォルト ルートを再配布することはありません。IS-IS にデフォルト ルートを生成し、ルート ポリシーでそのルートを制御できます。

IS-IS にインポートされたすべてのルートに使用する、デフォルトのメトリックも設定できます。

プレフィックスの抑制のリンク

デフォルトでは、IS-ISはシステムLSPの接続インターフェイスのアドレスをアドバタイズします。不要なインターフェイスアドレスのアドバタイズメントを抑制することで、LSPのサイズを削減し、IS-ISが維持するルートの数を削減して、コンバージェンス時間を短縮できます。

LSPのルート数を減らすために、次の2つのプレフィックス抑制方式が提供されています。

- グローバルレベルでは、他の接続されたプレフィックスを除く、パッシブインターフェイスに属するプレフィックスだけをアドバタイズするように選択できます。パッシブインターフェイスプレフィックスのみのアドバタイズ（305 ページ）を参照してください。
- インターフェイスレベルで、接続されたプレフィックスのアドバタイズメントを無効にできます。インターフェイスでのプレフィックスの抑制（306 ページ）を参照してください。

ロード バランシング

ロード バランシングを使用すると、ルータは、宛先アドレスから等距離内にあるすべてのルータのネットワーク ポートにトラフィックを分散できます。ロード バランシングは、ネットワークセグメントの使用率を向上させ、有効ネットワーク帯域幅を増加させます。

Cisco NX-OS は、ECMP（等コスト マルチパス）機能をサポートします。IS-IS ルート テーブルおよびユニキャスト RIB の等コスト パスは最大 16 です。これらのパスの一部または全部でトラフィックのロード バランシングが行われるように、IS-IS を設定できます。

BFD

この機能では、IPv4 および IPv6 用の双方向フォワーディング検出（BFD）をサポートします。BFD は、転送パスの障害を高速で検出することを目的にした検出プロトコルです。BFD は 2 台の隣接デバイス間のサブセカンド障害を検出し、BFD の負荷の一部を、サポートされるモジュール上のデータ プレーンに分散できるため、プロトコル hello メッセージよりも CPU を使いません。詳細については、『Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide』を参照してください。

仮想化のサポート

Cisco NX-OS は、IS-IS の複数のプロセス インスタンスをサポートします。各 IS-IS インスタンスは、システム制限まで複数の仮想ルーティングおよび転送（VRF）インスタンスをサポートできます。サポートされる IS-IS インスタンスの数については、『Cisco Nexus 9000 Series NX-OS Verified Scalability Guide』を参照してください。

高可用性およびグレースフル リスタート

Cisco NX-OS は、マルチレベルのハイ アベイラビリティ アーキテクチャを提供します。IS-IS は、ステートフル リスタートをサポートしています。これは、ノンストップ ルーティング（NSR）とも呼ばれます。IS-IS で問題が発生した場合は、以前の実行時状態からの再起動を試みます。この場合、ネイバーはいずれのネイバー イベントも登録しません。最初の再起動が正常ではなく、別の問題が発生した場合、RFC 3847 のとおり、IS-IS はグレースフル リスタートを試みます。グレースフル リスタート、つまり、Nonstop Forwarding（NSF）では、処理の再起動中も IS-IS がデータ転送パス上に存在し続けます。再起動中の IS-IS インターフェイスが稼働を再開すると、ネイバーを再探索して隣接関係を確立し、更新情報の送信を再開します。この時点で、NSF ヘルパーは、グレースフル リスタートが完了したと認識します。

ステートフル リスタートは次のシナリオで使用されます。

- プロセスでの問題発生後の最初の回復試行
- **system switchover** を使用したユーザ開始スイッチオーバー command

グレースフル リスタートは次のシナリオで使用されます。

- プロセスでの問題発生後の 2 回目の回復試行（4 分以内）
- **restart isis** を使用したプロセスの手動再起動 command
- アクティブ スーパーバイザの削除
- **reload module** *active-sup* コマンド

（注） グレースフル リスタートがデフォルトとなっており、ディセーブルにしないことを強く推奨します。

複数の IS-IS インスタンス

Cisco NX-OS は、同じノード上で動作する、IS-IS プロトコルの複数インスタンスをサポートしています。同一インターフェイスには複数のインスタンスを設定できません。すべてのインスタンスで同じシステム ルータ ID を使用します。サポートされる IS-IS インスタンスの数については、『Cisco Nexus 9000 Series NX-OS Verified Scalability Guide』を参照してください。

IS-IS の前提条件

IS-IS の前提条件は次のとおりです。

- IS-IS をイネーブルにする必要があります（「IS-IS 機能のイネーブル化」の項を参照）。

IS-IS に関する注意事項および制限事項

IS-IS 設定時の注意事項および制約事項は、次のとおりです。

- 明示的な設定がレベル 1/レベル 2 Cisco Nexus スイッチに追加されていない場合、IS-IS レベル 1 ルートは接続しているレベル 2 専用スイッチに入力されません。
- デフォルトの参照帯域幅が Cisco NX-OS と Cisco IOS では異なるため、アドバタイズされたトンネル IS-IS メトリックは、これら 2 つのオペレーティング システムによって異なります。
- すべての Cisco Nexus 9000 シリーズ スイッチと Cisco Nexus 3164Q および 31128PQ スイッチに対して、セグメント ルーティングを介した IS-IS を設定できます。詳細については、『Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide』を参照してください。

デフォルト設定

次の表に、IS-IS パラメータのデフォルト設定値を示します。

表 *21* : デフォルトの *IS-IS* パラメータ

パラメータ	デフォルト
アドミニストレーティブ ディスタンス	115
エリア レベル	Level-1-2
DIS プライオリティ	64
グレースフル リスタート	イネーブル
hello 乗数	3
hello パディング	イネーブル
hello タイム	10 秒
IS-IS 機能	ディセーブル
LSP 間隔	33
LSP MTU	1492
最大 LSP ライフタイム	1200 秒
最大パス	8
メトリック	40

パラメータ	デフォルト
参照帯域幅	40 Gbps

IS-IS の設定

IS-IS を設定する手順は、次のとおりです。

1. IS-IS 機能を有効にします（「IS-IS 機能の有効化」セクションを参照してください）。
2. IS-IS インスタンスを作成します（「IS-IS インスタンスの作成」セクションを参照してください）。
3. IS-IS インスタンスにインターフェイスを追加します（「インターフェイスでの IS-IS の設定」セクションを参照してください）。
4. 認証、メッシュ グループ、ダイナミック ホスト交換などのオプション機能を設定します。

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

IS-IS コンフィギュレーション モード

この項では、各コンフィギュレーション モードの開始方法について説明します。**?** コマンドを入力して、そのモードで利用可能なコマンドを表示できます。

ルータ コンフィギュレーション モード

次に、ルータ コンフィギュレーション モードを開始する例を示します。

```
switch#: configure terminal
switch(config)# router isis isp
switch(config-router)#
```

ルータ アドレス ファミリ コンフィギュレーション モード

次の例は、ネイバー アドレス ファミリ コンフィギュレーション モードの開始方法を示しています。

```
switch(config)# router isis isp
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)#
```

IS-IS 機能の有効化

IS-IS を設定する前に、IS-IS 機能を有効にする必要があります。

手順の概要

1. **configure terminal**
2. [**no**] **feature isis**
3. （任意） **show feature**
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	[**no**] **feature isis** 例： `switch(config)# feature isis`	IS-IS 機能を有効または無効にします。 このコマンドで **no** オプションを使用すると、IS-IS 機能を無効にし、関連付けられたすべての設定を削除します。
Step 3	（任意） **show feature** 例： `switch(config)# show feature`	有効および無効にされた機能を表示します。
Step 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

IS-IS インスタンスの作成

IS-IS インスタンスを作成し、そのインスタンスのエリア レベルを設定できます。

始める前に

IS-IS を有効にする必要があります（「IS-IS 機能の有効化」の項を参照）。

手順の概要

1. **configure terminal**
2. [**no**] **router isis** *instance-tag*
3. **net** *network-entity-title*
4. （任意） **is-type** {**level-1** | **level-2** | **level-1-2**}
5. （任意） **show isis** [**vrf** *vrf-name*] **process**
6. （任意） **distance** *value*
7. （任意） **log-adjacency-changes**
8. （任意） **lsp-mtu** *size*
9. （任意） **maximum-paths** *number*
10. （任意） **reference-bandwidth** *bandwidth-value* {**Mbps** | **Gbps**}
11. （任意） **clear isis** [*instance-tag*] **adjacency** [* | *system-id* | *interface*]
12. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	[**no**] **router isis** *instance-tag*
	例：
	`switch(config)# router isis Enterprise`
	`switch(config-router)#`
		instance tag を設定して、新しい IS-IS インスタンスを作成します。
		IS-IS インスタンスおよび関連するすべての設定を削除する場合は、このコマンドの **no** 形式を使用します。
		（注）
		IS-IS インスタンスに関するすべての設定を完全に削除するには、インターフェイス モードで設定した IS-IS コマンドも削除する必要があります。

Step 3	**net** *network-entity-title* 例： `switch(config-router)# net 47.0004.004d.0001.0001.0c11.1111.00`	この IS-IS インスタンスに対応する NET を設定します。
Step 4	（任意） **is-type** {**level-1**	**level-2**	**level-1-2**} 例： `switch(config-router)# is-type level-2`	この IS-IS インスタンスのエリア レベルを設定します。デフォルトは level-1-2 です。

	コマンドまたはアクション	目的
Step 5	（任意） **show isis** [**vrf** *vrf-name*] **process** 例： `switch(config-router)# show isis process`	すべての IS-IS インスタンスについて、IS-IS 要約情報を表示します。
Step 6	（任意） **distance** *value* 例： `switch(config-router)# distance 30`	IS-IS のアドミニストレーティブ ディスタンスを設定します。範囲は 1 ～ 255 です。デフォルトは 115 です。
Step 7	（任意） **log-adjacency-changes** 例： `switch(config-router)# log-adjacency-changes`	IS-IS ネイバーのステートが変化するたびに、システム メッセージを送信します。
Step 8	（任意） **lsp-mtu** *size* 例： `switch(config-router)# lsp-mtu 600`	この IS-IS インスタンスにおける LSP の MTU を設定します。指定できる範囲は 128 ～ 4352 バイトです。デフォルトは 1492 です。
Step 9	（任意） **maximum-paths** *number* 例： `switch(config-router)# maximum-paths 6`	IS-IS がルート テーブルで維持する等コスト パスの最大数を設定します。範囲は 1 ～ 64 です。デフォルト値は 8 です。
Step 10	（任意） **reference-bandwidth** *bandwidth-value* {**Mbps** | **Gbps**} 例： `switch(config-router)# reference-bandwidth 100 Gbps`	IS-IS コスト メトリックの計算に使用する、デフォルトの基準帯域幅を設定します。指定できる範囲は 1 ～ 4000 Gbps です。デフォルトは 40 Gbps です。
Step 11	（任意） **clear isis** [*instance-tag*] **adjacency** [* | *system-id* | *interface*] 例： `switch(config-router)# clear isis adjacency *`	ネイバーの統計情報を消去し、この IS-IS インスタンスの隣接関係を削除します。
Step 12	（任意） **copy running-config startup-config** 例： `switch(config-router)# copy running-config startup-config`	この設定変更を保存します。

例

レベル 2 エリアで IS-IS インスタンスを作成する例を示します。

```
switch# configure terminal
switch(config)# router isis Enterprise
switch(config-router)# net 47.0004.004d.0001.0001.0c11.1111.00
```

```
switch(config-router)# is-type level-2
switch(config-router)# copy running-config startup-config
```

IS-IS インスタンスの再起動

IS-ISインスタンスを再起動できます。この処理では、インスタンスのすべてのネイバーが消去されます。

IS-ISインスタンスを再起動し、関連付けられたすべてのネイバーを削除するには、次のコマンドを使用します。

手順の概要

1. **restart isis** *instance-tag*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**restart isis** *instance-tag* 例: `switch(config)# restart isis Enterprise`	IS-IS インスタンスを再起動し、すべてのネイバーを削除します。

IS-IS のシャットダウン

IS-IS インスタンスをシャットダウンできます。シャットダウンすると、その IS-IS インスタンスがディセーブルになり、設定が保持されます。

IS-IS インスタンスをシャットダウンするには、ルータ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **shutdown**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**shutdown** 例: `switch(config-router)# shutdown`	IS-IS インスタンスをディセーブルにします。

インターフェイスでの IS-IS の設定

IS-IS インスタンスにインターフェイスを追加できます。

始める前に

IS-IS を有効にする必要があります（「IS-IS 機能の有効化」の項を参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. （任意） **medium** {**broadcast** | **p2p**}
4. {**ip** | **ipv6**} **router isis** *instance-tag*
5. （任意） **show isis** [**vrf** *vrf-name*] [*instance-tag*] **interface** [*interface-type slot/port*]
6. （任意） **isis circuit-type** {**level-1** | **level-2** | **level-1-2**}
7. （任意） **isis metric** *value* {**level-1** | **level-2**}
8. （任意） **isis passive** {**level-1** | **level-2** | **level-1-2**}
9. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	（任意） **medium** {**broadcast**	**p2p**} 例： `switch(config-if)# medium p2p`	インターフェイスにブロードキャストモードまたはポイントツーポイントモードを設定します。IS-IS はこのモードを継承します。
Step 4	{**ip**	**ipv6**} **router isis** *instance-tag* 例： `switch(config-if)# ip router isis Enterprise`	この IPv4 または IPv6 インターフェイスを IS-IS インスタンスに関連付けます。

	コマンドまたはアクション	目的
Step 5	（任意） **show isis** [**vrf** *vrf-name*] [*instance-tag*] **interface** [*interface-type slot/port*] 例： `switch(config-if)# show isis Enterprise ethernet 1/2`	インターフェイスの IS-IS 情報を表示します。
Step 6	（任意） **isis circuit-type** {**level-1** | **level-2** | **level-1-2**} 例： `switch(config-if)# isis circuit-type level-2`	このインターフェイスが参加する隣接関係のタイプを設定します。このコマンドを使用するのは、レベル 1 とレベル 2 の両方のエリアにルータが関係する場合だけです。
Step 7	（任意） **isis metric** *value* {**level-1** | **level-2**} 例： `switch(config-if)# isis metric 30`	このインターフェイスの IS-IS メトリックを設定します。指定できる範囲は 1 ～ 16777214 です。デフォルトは 10 です。
Step 8	（任意） **isis passive** {**level-1** | **level-2** | **level-1-2**} 例： `switch(config-if)# isis passive level-2`	インターフェイスが隣接関係を形成しないようにしながら、なおかつ、インターフェイスに関連付けられたプレフィックスをアドバタイズするようにします。
Step 9	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

例

次に、IS-IS インスタンスに Ethernet 1/2 インターフェイスを追加する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ip router isis Enterprise
switch(config-if)# copy running-config startup-config
```

インターフェイスでの IS-IS のシャットダウン

インターフェイス上で IS-IS を正常にシャットダウンできます。これにより、すべての隣接関係が削除され、このインターフェイスで IS-IS トラフィックが停止しますが、IS-IS 設定は保持されます。

インターフェイス上で IS-IS を無効にするには、インターフェイス設定モードで次のコマンドを使用します。

手順の概要

1. **isis shutdown**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**isis shutdown** 例： `switch(config-if)# isis shutdown`	このインターフェイスでIS-ISを無効にします。IS-IS インターフェイスの設定は保持されます。

エリアでの IS-IS 認証の設定

エリアで LSP を認証するように IS-IS を設定できます。

始める前に

IS-IS を有効にする必要があります。「IS-IS 機能の有効化」を参照してください。

キーチェーンをIS-IS設定から参照する場合は、グローバル設定モードでキーチェーンを設定する必要があります。キーチェーン管理の詳細については、「Cisco Nexus 9000 シリーズ NX-OS セキュリティ設定ガイド」を参照してください。

手順の概要

1. **configure terminal**
2. **router isis** *instance-tag*
3. **authentication-type** {**cleartext** | **md5**} {**level-1** | **level-2**}
4. **authentication key-chain** *key* {**level-1** | **level-2**}
5. （任意） **authentication-check** {**level-1** | **level-2**}
6. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

	コマンドまたはアクション	目的

Step 2	**router isis** *instance-tag*
	例：
	`switch(config)# router isis Enterprise`
	`switch(config-router)#`
		instance tag を設定して、新しい IS-IS インスタンスを作成します。

Step 3	**authentication-type** {**cleartext** | **md5**} {**level-1** | **level-2**} 例： `switch(config-router)# authentication-type cleartext level-2`	クリアテキストまたは MD5 認証ダイジェストとして、レベル 1 またはレベル 2 エリアに使用する認証方式を設定します。
Step 4	**authentication key-chain** *key* {**level-1** | **level-2**} 例： `switch(config-router)# authentication key-chain ISISKey level-2`	IS-IS エリア レベル認証に使用する認証キーを設定します。
Step 5	（任意） **authentication-check** {**level-1** | **level-2**} 例： `switch(config-router)# authentication-check level-2`	受信パケットの認証パラメータ チェックを有効にします。
Step 6	（任意） **copy running-config startup-config** 例： `switch(config-router)# copy running-config startup-config`	この設定変更を保存します。

例

IS-IS インスタンスにクリアテキスト認証を設定する例を示します。

```
switch# configure terminal
switch(config)# router isis Enterprise
switch(config-router)# authentication-type cleartext level-2
switch(config-router)# authentication key-chain ISISKey level-2
switch(config-router)# copy running-config startup-config
```

インターフェイスでの IS-IS 認証の設定

インターフェイスで Hello パケットを認証するように IS-IS を設定できます。

始める前に

IS-IS を有効にする必要があります（「IS-IS 機能の有効化」の項を参照）。

手順の概要

1. **configure terminal**

2. **interface** *interface-type slot/port*
3. **isis authentication-type** {**cleartext** | **md5**} {**level-1** | **level-2**}
4. **isis authentication key-chain** *key* {**level-1** | **level-2**}
5. （任意） **isis authentication-check** {**level-1** | **level-2**}
6. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**isis authentication-type** {**cleartext** | **md5**} {**level-1** | **level-2**} 例： `switch(config-if)# isis authentication-type cleartext level-2`	クリアテキストまたは MD5 認証ダイジェストとして、このインターフェイスにおける IS-IS 認証タイプを設定します。
Step 4	**isis authentication key-chain** *key* {**level-1** | **level-2**} 例： `switch(config-if)# isis authentication-key ISISKey level-2`	このインターフェイス上で IS-IS に使用する認証キーを設定します。
Step 5	（任意） **isis authentication-check** {**level-1** | **level-2**} 例： `switch(config-if)# isis authentication-check`	受信パケットの認証パラメータ チェックを有効にします。
Step 6	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

例

IS-IS インスタンスにクリアテキスト認証を設定する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# isis authentication-type cleartext level-2
switch(config-if)# isis authentication key-chain ISISKey
switch(config-if)# copy running-config startup-config
```

メッシュ グループの設定

メッシュ グループにインターフェイスを追加することによって、そのメッシュ グループ内のインターフェイスに対する LSP フラッディングの量を制限できます。任意で、メッシュ グループ内のインターフェイスに対して、すべての LSP フラッディングをブロックすることもできます。

メッシュ グループにインターフェイスを追加するには、インターフェイス設定モードで次のコマンドを使用します。

手順の概要

1. **isis mesh-group** {**blocked** | *mesh-id*}

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**isis mesh-group** {**blocked** | *mesh-id*} 例: `switch(config-if)# isis mesh-group 1`	メッシュ グループにこのインターフェイスを追加します。範囲は 1 ～ 4294967295 です。

指定中継システムの設定

インターフェイスプライオリティを設定することによって、ルータがマルチアクセスネットワークの代表中継システム（DIS）になるように設定できます。

DIS を設定するには、インターフェイス設定モードで次のコマンドを使用します。

手順の概要

1. **isis priority** *number* {**level-1** | **level-2**}

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**isis priority** *number* {**level-1** | **level-2**} 例： `switch(config-if)# isis priority 100 level-1`	DIS 選択のためのプライオリティを設定します。指定できる範囲は 0 ～ 127 です。デフォルトは 64 です。

ダイナミック ホスト交換の設定

ダイナミック ホスト交換を使用してシステム ID とルータのホスト名をマッピングするように、IS-IS を設定できます。

ダイナミック ホスト交換を設定するには、ルータ設定モードで次のコマンドを使用します。

手順の概要

1. **hostname dynamic**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**hostname dynamic** 例： `switch(config-router)# hostname dynamic`	ダイナミック ホスト交換をイネーブルにします。

過負荷ビットの設定

最短パス優先（SPF）の計算で中間ホップとしてこのルータを使用しないことを他のルータに通知するように、ルータを設定できます。任意で、起動時にBGPがコンバージェンスするまで、一時的に過負荷ビットを設定することもできます。

過負荷ビットを設定する以外に、レベル 1 またはレベル 2 トラフィックに関して、LSP からの特定タイプの IP プレフィックス アドバタイズメントを抑制することが必要な場合もあります。

過負荷ビットを設定するには、ルータ コンフィギュレーションモードで次のコマンドを使用します。

手順の概要

1. **set-overload-bit** {**always** | **on-startup** {*seconds* | **wait-for bgp** *as-number*}} [**suppress** [**interlevel** | **external**]]

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**set-overload-bit** {**always**	**on-startup** {*seconds*	**wait-for bgp** *as-number*}} [**suppress** [**interlevel**	**external**]] 例： `switch(config-router)# set-overload-bit on-startup 30`	IS-IS に過負荷ビットを設定します。*seconds* の範囲は 5 ～ 86400 です。

接続ビットの設定

Attached ビットを設定すると、レベル 1 ルータがレベル 2 エリアへのデフォルト ルートとして使用するレベル 1/レベル 2 ルータを制御できます。Attached ビットの設定をディセーブルにすると、レベル 1 ルータはこのレベル 1/レベル 2 ルータを使用してレベル 2 エリアに接続しなくなります。

レベル 1/レベル 2 ルータの Attached ビットを設定するには、ルータ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. [**no**] **set-attached-bit**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	[**no**] **set-attached-bit** 例： `switch(config-router)# no attached-bit`	Attached ビットを設定するようにレベル 1/レベル 2 ルータを設定します。この機能は、デフォルトでイネーブルにされています。

hello パディングの一時モードの設定

hello パディングの一時モードを設定すると、IS-IS が隣接関係を確立するときに hello パケットをパディングし、IS-IS が隣接関係を確立したあとでそのパディングを削除できます。

hello パディングのモードを設定するには、ルータ設定モードで次のコマンドを使用します。

手順の概要

1. [**no**] **isis hello-padding**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	[**no**] **isis hello-padding** 例： `switch(config-if)# no isis hello-padding`	完全な最大伝送単位（MTU）に hello パケットをパディングします。デフォルトではイネーブルになっています。パディングの一時モードを設定するには、このコマンドの **no** 形式を使用します。

サマリー アドレスの設定

ルーティング テーブルでサマリー アドレスによって表されるサマリ アドレスを作成できます。1 つのサマリーアドレスに、特定のレベルのアドレス グループを複数含めることができます。Cisco NX-OS は固有性の強いすべてのルートのうち、最小メトリックをアドバタイズします。

始める前に

IS-IS を有効にする必要があります（「IS-IS 機能の有効化」の項を参照）。

手順の概要

1. **configure terminal**
2. **router isis** *instance-tag*
3. **address-family** {**ipv4** | **ipv6**} **unicast**
4. **summary-address** *ip-prefix/mask-len* {**level-1 | level-2 | level-1-2**}
5. （任意） **show isis** [**vrf***vrf-name*] {**ip** | **ipv6**} **summary-address** *ip-prefix* [**longer-prefixes**]
6. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的

Step 2	**router isis** *instance-tag*
	例：
	`switch(config)# router isis Enterprise`
	`switch(config-router)#`
		instance tag を設定して、新しい IS-IS インスタンスを作成します。

Step 3	**address-family** {**ipv4** | **ipv6**} **unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		アドレス ファミリ設定モードを開始します。

Step 4	**summary-address** *ip-prefix/mask-len* {**level-1 | level-2 | level-1-2**} 例： `switch(config-router-af)# summary-address 192.0.2.0/24 level-2`	IPv4 アドレスまたは IPv6 アドレスに対応する、IS-IS エリア用のサマリー アドレスを設定します。

Step 5	（任意） **show isis** [**vrf***vrf-name*] {**ip** | **ipv6**} **summary-address** *ip-prefix* [**longer-prefixes**]
	例：
	`Example:`
	`switch(config-router-af)# show isis ip summary-address`
		IS-IS IPv4 または IPv6 サマリー アドレス情報を表示します。

Step 6	（任意） **copy running-config startup-config** 例： `switch(config-router-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、IS-IS の IPv4 ユニキャスト サマリー アドレスを設定する例を示します。

```
switch# configure terminal
switch(config)# router isis Enterprise
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# summary-address 192.0.2.0/24 level-2
switch(config-router-af)# copy running-config startup-config
```

再配布の設定

別のルーティング プロトコルからのルーティング情報を受け入れて、IS-IS ネットワークを通じてその情報を再配布するように、IS-IS を設定できます。任意で、再配布ルートのためのデフォルトルートを割り当てることができます。

始める前に

IS-IS を有効にする必要があります（「IS-IS 機能の有効化」の項を参照）。

手順の概要

1. **configure terminal**
2. **router isis** *instance-tag*
3. **address-family** {**ipv4** | **ipv6**} **unicast**
4. **redistribute** {**bgp** *as* | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static** | **direct**} **route-map** *map-name*
5. （任意） **default-information originate** [**always**] [**route-map** *map-name*]
6. （任意） **distribute** {**level-1** | **level-2**} **into** {**level-1** | **level-2**} {**route-map** *route-map* | **all**}
7. （任意） **show isis** [**vrf** *vrf-name*] {**ip** | **ipv6**} **route** *ip-prefix* [*detail* | **longer-prefixes** [**summary** | **detail**]]
8. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router isis** *instance-tag*
	例：
	`switch(config)# router isis Enterprise`
	`switch(config-router)#`
		instance tag を設定して、新しい IS-IS インスタンスを作成します。

Step 3	**address-family** {**ipv4**	**ipv6**} **unicast**
		例：
		`switch(config-router)# address-family ipv4 unicast`
		`switch(config-router-af)#`
			アドレス ファミリ設定モードを開始します。

Step 4	**redistribute** {**bgp** *as*	{**eigrp**	**isis**	**ospf**	**ospfv3**	**rip**} *instance-tag*	**static**	**direct**} **route-map** *map-name* 例： `switch(config-router-af)# redistribute eigrp 201 route-map ISISmap`	他のプロトコルからのルートを IS-IS に再配布します。
Step 5	（任意） **default-information originate** [**always**] [**route-map** *map-name*] 例：	IS-IS へのデフォルト ルートを生成します。

	コマンドまたはアクション	目的
	`switch(config-router-af)# default-information originate always`	
Step 6	（任意） **distribute** {**level-1** | **level-2**} **into** {**level-1** | **level-2**} {**route-map** *route-map* | **all**} 例： `switch(config-router-af)# distribute level-1 into level-2 all`	一方の IS-IS レベルから他方の IS-IS レベルへ、ルートを再配布します。
Step 7	（任意） **show isis** [**vrf** *vrf-name*] {**ip** | **ipv6**} **route** *ip-prefix* [*detail* | **longer-prefixes** [**summary** | **detail**]] 例： `switch(config-router-af)# show isis ip route`	IS-IS ルートを表示します。
Step 8	（任意） **copy running-config startup-config** 例： `switch(config-router-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、EIGRP を IS-IS に再配布する例を示します。

```
switch# configure terminal
switch(config)# router isis Enterprise
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# redistribute eigrp 201 route-map ISISmap
switch(config-router-af)# copy running-config startup-config
```

再配布されるルート数の制限

ルートの再配布によって、IS-IS ルートテーブルに多くのルートが追加される可能性があります。外部プロトコルから受け取るルートの数の上限を設定できます。IS-IS には、再配布ルートの制限を設定するために次のオプションが用意されています。

- 上限固定：IS-IS が設定された最大値に達すると、メッセージをログに記録します。IS-IS は以降の再配布ルートを受け取りません。任意で、最大値のしきい値パーセンテージを設定して、IS-IS がこのしきい値を超えたときに警告を記録するようにすることもできます。
- 警告のみ：IS-IS が最大値に達したときのみ、警告のログを記録します。IS-IS は引き続き再配布ルートを受け取ります。
- 取り消し：IS-IS が最大値に達したときにタイムアウト期間を開始します。タイムアウト期間の経過後、現在の再配布ルートの数が最大制限より少ない場合、IS-IS はすべての再配布ルートを要求します。現在の再配布ルートの数が最大制限に達している場合、IS-IS はすべての再

配布ルートを取り消します。IS-IS が以降の再配布ルートを受け取るには、この状態を解消する必要があります。任意で、タイムアウト期間を設定できます。

始める前に

IS-IS を有効にする必要があります。

手順の概要

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

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**router isis** *instance-tag*
	例：
	`switch(config)# router isis Enterprise`
	`switch(config-router)#`
		instance tag を設定して、新しい IS-IS インスタンスを作成します。

Step 3	**redistribute** {**bgp** *id* | **direct** | **eigrp***id* | **isis** *id* | **ospf** *id* | **rip** *id* | **static**} **route-map** *map-name* 例： `switch(config-router)# redistribute bgp route-map FilterExternalBGP`	設定したルートマップ経由で、選択したプロトコルを IS-IS に再配布します。
Step 4	**redistribute maximum-prefix** *max* [*threshold*] [**warning-only** | **withdraw** [*num-retries timeout*]] 例： `switch(config-router)# redistribute maximum-prefix 1000 75 warning-only`	IS-IS が配布するプレフィックスの最大数を指定します。有効な範囲は 1 ～ 65535 です。次の項目を任意で指定できます。 • *threshold*：警告メッセージをトリガーする最大プレフィックスの割合。 • **warning-only**：プレフィックスの最大数を超えた場合に警告メッセージを記録します。

	コマンドまたはアクション	目的
		• **withdraw**：再配布されたすべてのルートを取り消します。オプション選択で、再配布されたルートの取得を試みることができます。*num-retries* の範囲は 1 ～ 12 です。*timeout* は 60 ～ 600 秒です。デフォルトは 300 秒です。**clear isis redistribution** コマンドは、すべてのルートが取り消された場合に使用します。
Step 5	（任意） **show running-config isis** 例： `switch(config-router)# show running-config isis`	IS-IS の設定を表示します。
Step 6	（任意） **copy running-config startup-config** 例： `switch(config-router)# copy running-config startup-config`	この設定変更を保存します。

例

次に、IS-IS に再配布されるルートの数を制限する例を示します。

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

パッシブインターフェイスプレフィックスのみのアドバタイズ

パッシブインターフェイスに属するプレフィックスだけがシステムリンクステートパケット（LSP）でアドバタイズされるように指定できます。

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例： `switch# configure terminal switch(config)#`	グローバルコンフィギュレーションモードを開始します。
Step 2	**router isis** *instance-tag* 例：	instance tag を設定して、新しい IS-IS インスタンスを作成します。

	コマンドまたはアクション	目的
	`switch(config)# router isis 200` `switch(config-router)#`	

Step 3	**address-family** {**ipv4** | **ipv6**} **unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		アドレス ファミリ設定モードを開始します。

Step 4	**[no] advertise passive-only {level-1 | level-2}**
	例：
	`switch(config-router-af)# advertise passive-only level-1`
	`switch(config-router-af)#`
		パッシブインターフェイスに属するプレフィックスのみのアドバタイズメントをイネーブルにします。

例

次に、パッシブインターフェイスに属するプレフィックスのアドバタイズのみをイネーブルにする例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# address-family ipv4 unicast
switch(config-router-af)# advertise passive-only level-1
```

インターフェイスでのプレフィックスの抑制

IS-IS インターフェイスが システム リンクステート パケット（LSP） 内の接続されたプレフィックスをアドバタイズせずに隣接の形成に参加できるようにします。

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

	コマンドまたはアクション	目的

Step 3	**[no] isis suppress**
	例：
	`switch(config-if)# isis suppress`
	`switch(config-if)#`
		インターフェイスで接続されているプレフィックスのアドバタイズメントを無効にします。

例

次に、システムリンクステートパケット（LSP）でインターフェイスの接続されたプレフィックスのアドバタイズを抑制する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# isis suppress
```

厳密な隣接モードのディセーブル化

IPv4 と IPv6 の両方のアドレス ファミリがイネーブルの場合、厳格な隣接モードはデフォルトでイネーブルです。このモードでは、デバイスが両方のアドレス ファミリにイネーブルでない任意のルータとの隣接関係を形成しません。厳格な隣接モードは、**no adjacency-check** コマンドを使用してディセーブルにできます。 コマンドを使用する必要があります。

始める前に

IS-IS を有効にする必要があります（「IS-IS 機能の有効化」の項を参照）。

手順の概要

1. **configure terminal**
2. **router isis** *instance-tag*
3. **address-family ipv4 unicast**
4. **no adjacency-check**
5. **exit**
6. **address-family ipv6 unicast**
7. **no adjacency-check**
8. （任意） **show running-config isis**
9. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**router isis** *instance-tag*
	例：
	`switch(config)# router isis Enterprise`
	`switch(config-router)#`
		instance tag を設定して、新しい IS-IS インスタンスを作成します。

Step 3	**address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		アドレスファミリコンフィギュレーションモードを開始します。

Step 4	**no adjacency-check** 例： `switch(config-router-af)# no adjacency-check`	IPv4 アドレスファミリに関する厳格な隣接モードをディセーブルにします。

Step 5	**exit**
	例：
	`switch(config-router-af)# exit`
	`switch(config-router)#`
		アドレスファミリコンフィギュレーションモードを終了します。

Step 6	**address-family ipv6 unicast**
	例：
	`switch(config-router)# address-family ipv6 unicast`
	`switch(config-router-af)#`
		アドレスファミリコンフィギュレーションモードを開始します。

Step 7	**no adjacency-check** 例： `switch(config-router-af)# no adjacency-check`	IPv6 アドレスファミリに関する厳格な隣接モードをディセーブルにします。
Step 8	（任意） **show running-config isis** 例： `switch(config-router-af)# show running-config isis`	IS-IS の設定を表示します。

	コマンドまたはアクション	目的
Step 9	（任意） **copy running-config startup-config** 例： `switch(config-router-af)# copy running-config startup-config`	この設定変更を保存します。

グレースフル リスタートの設定

IS-IS のグレースフル リスタートを設定できます。

始める前に

IS-IS を有効にする必要があります（「IS-IS 機能の有効化」の項を参照）。

手順の概要

1. **configure terminal**
2. **router isis** *instance-tag*
3. **graceful restart**
4. **graceful-restart t3 manual** *time*
5. （任意） **show running-config isis**
6. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router isis** *instance-tag*
	例：
	`switch(config)# router isis Enterprise`
	`switch(config-router)#`
		名前を設定して、新しい IS-IS プロセスを作成します。

Step 3	**graceful restart** 例： `switch(config-router)# graceful-restart`	グレースフル リスタートおよびグレースフル リスタート ヘルパー機能を有効にします。デフォルトでは、有効です。

	コマンドまたはアクション	目的
Step 4	**graceful-restart t3 manual** *time* 例： `switch(config-router)# graceful-restart t3 manual 300`	グレースフル リスタート T3 タイマーを設定します。有効な範囲は 30 ～ 65535 秒です。デフォルトは 60 です。
Step 5	（任意） **show running-config isis** 例： `switch(config-router)# show running-config isis`	IS-IS の設定を表示します。
Step 6	（任意） **copy running-config startup-config** 例： `switch(config-router)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、グレースフル リスタートを有効にする例を示します。

```
switch# configure terminal
switch(config)# router isis Enterprise
switch(config-router)# graceful-restart
switch(config-router)# copy running-config startup-config
```

仮想化の設定

複数の IS-IS インスタンスと複数の VRF を設定できます。また、各 VRF で同じまたは複数の IS-IS インスタンスを使用することもできます。VRF に IS-IS インターフェイスを割り当てます。

設定した VRF に NET を設定する必要があります。

（注） インターフェイスの VRF を設定した後に、インターフェイスの他のすべてのパラメータを設定します。インターフェイスの VRF を設定すると、そのインターフェイスのすべての設定が削除されます。

始める前に

IS-IS を有効にする必要があります（「IS-IS 機能の有効化」の項を参照）。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf-name*

3. **exit**
4. **router isis** *instance-tag*
5. （任意） **vrf** *vrf-name*
6. **net** *network-entity-title*
7. **exit**
8. **exit**
9. **interface ethernet** *slot/port*
10. **vrf member** *vrf-name*
11. {**ip** | **ipv6**) **address** *ip-prefix/length*
12. {**ip** | **ipv6**) **router isis** *instance-tag*
13. （任意） **show isis** [**vrf** *vrf-name*] [*instance-tag*] **interface** [*interface-type slot/port*]
14. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context RemoteOfficeVRF`
	`switch(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。

Step 3	**exit**
	例：
	`switch(config-vrf)# exit`
	`switch(config)#`
		VRF設定モードを終了します。

Step 4	**router isis** *instance-tag*
	例：
	`switch(config)# router isis Enterprise`
	`switch(config-router)#`
		instance tag を設定して、新しい IS-IS インスタンスを作成します。

Step 5	（任意） **vrf** *vrf-name*
	例：
	`switch(config-router)# vrf RemoteOfficeVRF`
	`switch(config-router-vrf)#`
		ルータ VRF 設定モードを開始します。

	コマンドまたはアクション	目的
Step 6	**net** *network-entity-title* 例： `switch(config-router-vrf)# net 47.0004.004d.0001.0001.0c11.1111.00`	この IS-IS インスタンスに対応する NET を設定します。
Step 7	**exit** 例： `switch(config-router-vrf)# exit switch(config-router)#`	ルータ VRF 設定モードを終了します。
Step 8	**exit** 例： `switch(config-router)# exit switch(config)#`	ルータ設定モードを終了します。
Step 9	**interface ethernet** *slot/port* 例： `switch(config)# interface ethernet 1/2 switch(config-if)#`	インターフェイス設定モードを開始します。
Step 10	**vrf member** *vrf-name* 例： `switch(config-if)# vrf member RemoteOfficeVRF`	このインターフェイスを VRF に追加します。
Step 11	{**ip** | **ipv6**) **address** *ip-prefix/length* 例： `switch(config-if)# ip address 192.0.2.1/16`	このインターフェイスの IP アドレスを設定します。このステップは、このインターフェイスを VRF に割り当てたあとに行う必要があります。
Step 12	{**ip** | **ipv6**) **router isis** *instance-tag* 例： `switch(config-if)# ip router isis Enterprise`	この IPv4 または IPv6 インターフェイスを IS-IS インスタンスに関連付けます。
Step 13	（任意） **show isis** [**vrf** *vrf-name*] [*instance-tag*] **interface** [*interface-type slot/port*] 例： `switch(config-if)# show isis Enterprise ethernet 1/2`	VRF のインターフェイスに関する IS-IS 情報を表示します。
Step 14	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

例

次に、VRF を作成して、その VRF にインターフェイスを追加する例を示します。

```
switch# configure terminal
switch(config)# vrf context NewVRF
switch(config-vrf)# exit
switch(config)# router isis Enterprise
switch(config-router)# vrf NewVRF
switch(config-router-vrf)# net 47.0004.004d.0001.0001.0c11.1111.00
switch(config-router-vrf)# exit
switch(config-router)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# vrf member NewVRF
switch(config-if)# ip address 192.0.2.1/16
switch(config-if)# ip router isis Enterprise
switch(config-if)# copy running-config startup-config
```

IS-IS の調整

ネットワーク要件に合わせて IS-IS を調整できます。

IS-IS を調整するには、次のオプション コマンドを使用します。

手順の概要

1. （任意） **lsp-gen-interval** [**level-1** | **level-2**] *lsp-max-wait* [*lsp-initial-wait lsp-second-wait*]
2. （任意） **max-lsp-lifetime** ライフタイム
3. （任意） **metric-style transition**
4. （任意） **spf-interval** [**level-1** | **level-2**] *spf-max-wait* [*spf-initial-wait spf-second-wait*]
5. （任意） **adjacency-check**
6. （任意） **isis csnp-interval** *seconds* [**level-1** | **level-2**]
7. （任意） **isis hello-interval** *seconds* [**level-1** | **level-2**]
8. （任意） **isis hello-multiplier** *num* [**level-1** | **level-2**]
9. （任意） **isis lsp-interval** *milliseconds*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	（任意） **lsp-gen-interval** [**level-1** | **level-2**] *lsp-max-wait* [*lsp-initial-wait lsp-second-wait*] 例： `switch(config-router)# lsp-gen-interval level-1 500 500 500`	LSP 発生に関する IS-IS スロットルを設定します。オプション パラメータは次のとおりです。 • *lsp-max-wait*：トリガーから LSP 発生までの最大待ち時間。指定できる範囲は 500 ～ 65535 ミリ秒です。

	コマンドまたはアクション	目的
		• *lsp-initial-wait*：トリガーから LSP 発生までの初期待ち時間。指定できる範囲は 50 ～ 65535 ミリ秒です。 • *lsp-second-wait*：バックオフ時の LSP スロットルに使用する第 2 待ち時間。指定できる範囲は 50 ～ 65535 ミリ秒です。
Step 2	（任意） **max-lsp-lifetime** ライフタイム 例： `switch(config-router)# max-lsp-lifetime 500`	LSP の最大ライフタイムを秒数で設定します。有効な範囲は 1 ～ 65535 です。デフォルトは 1200 です。
Step 3	（任意） **metric-style transition** 例： `switch(config-router)# metric-style transition`	IS-IS がナロー メトリック スタイルのタイプ、長さ、値（TLV）オブジェクトとワイドメトリック スタイルの TLV オブジェクトの両方を生成して受け取ることができるようにします。デフォルトではディセーブルになっています。
Step 4	（任意） **spf-interval** [**level-1** | **level-2**] *spf-max-wait* [*spf-initial-wait spf-second-wait*] 例： `switch(config-router)# spf-interval level-2 500 500 500`	LSA 到着までのインターバルを設定します。オプション パラメータは次のとおりです。 • *lsp-max-wait*：トリガーから SPF 計算までの最大待ち時間。指定できる範囲は 500 ～ 65535 ミリ秒です。 • *lsp-initial-wait*：トリガーから SPF 計算までの初期待ち時間。指定できる範囲は 50 ～ 65535 ミリ秒です。 • *lsp*-second-wait：バックオフ時の SPF 計算に使用する第 2 待ち時間。指定できる範囲は 50 ～ 65535 ミリ秒です。
Step 5	（任意） **adjacency-check** 例： `switch(config-router-af)# adjacency-check`	隣接関係チェックを実行し、IS-IS インスタンスが同じアドレス ファミリをサポートするリモート IS-IS エンティティに限って隣接関係を形成していることを確認します。このコマンドは、デフォルトでイネーブルになっています。
Step 6	（任意） **isis csnp-interval** *seconds* [**level-1** | **level-2**] 例： `switch(config-if)# isis csnp-interval 20`	IS-IS に Complete Sequence Number PDU（CNSP）インターバルを秒数で設定します。有効な範囲は 1 ～ 65535 です。デフォルトは 10 です。
Step 7	（任意） **isis hello-interval** *seconds* [**level-1** | **level-2**] 例：	IS-IS に hello 間隔を秒数で設定します。有効な範囲は 1 ～ 65535 です。デフォルトは 10 です。

	コマンドまたはアクション	目的
	`switch(config-if)# isis hello-interval 20`	
Step 8	（任意） **isis hello-multiplier** *num* [**level-1** | **level-2**] 例： `switch(config-if)# isis hello-multiplier 20`	ルータが隣接関係を破棄するまでに、ネイバーが見逃さなければならない IS-IS hello パケットの数を指定します。指定できる範囲は 3 ～ 1000 です。デフォルトは 3 です。
Step 9	（任意） **isis lsp-interval** *milliseconds* 例： `switch(config-if)# isis lsp-interval 20`	フラッディング時にこのインターフェイスで LSP が送信される間隔をミリ秒数で設定します。指定できる範囲は 10 ～ 65535 です。デフォルトは 33 です。

IS-IS 設定の確認

IS-IS の設定を表示するには、次のいずれかの作業を行います。

コマンド	目的
show isis [*instance-tag*] **adjacency** [*interface*] [**detail** | **summary**] [**vrf** *vrf-name*]	IS-IS の隣接関係を表示します。**clear isis adjacency** コマンドを使用して、これらの統計情報をクリアします。 （注） ホスト名が 14 文字未満の場合、**show isis adjacency** コマンドはホスト名を表示します。それ以外の場合は、システム ID が表示されます。
show isis [*instance-tag*] **database** [**level-1** | **level-2**] [**detail** | **summary**] [*lsp-id*] [{**ip** | **ipv6**} **prefix***ip-prefix*] | [**router-id** *router-id*] | [**adjacency** *node-id*] | [**zero-sequence**]} [**vrf** *vrf-name*]	IS-IS LSP データベースを表示します。
show isis [*instance-tag*] *hostname* [**vrf** *vrf-name*]	ダイナミック ホスト交換情報を表示します。
show isis [*instance-tag*] **interface** [**brief** | *interface*] [**level-1 | level-2**] [**vrf***vrf-name*]	IS-IS インターフェイス情報を表示します。
show isis [*instance-tag*] **mesh-group** [*mesh-id*] [**vrf***vrf-name*]	メッシュ グループ情報を表示します。
show isis [*instance-tag*] **protocol** [**vrf** *vrf-name*]	IS-IS プロトコルに関する情報を表示します。
show isis [*instance-tag*] {**ip** | **ipv6**} **redistribute route** [*ip-address* | **summary**] [*ip-prefix*] [**longer-prefixes** [**summary**]] [**vrf** *vrf-name*]	IS-IS のルート再配布情報を表示します。

コマンド	目的
show isis [*instance-tag*] {**ip** | **ipv6**} **route** [*ip-address* | **summary**] [*ip-prefix*] [**longer-prefixes** [**summary**]] [**detail**] [**vrf** *vrf-name*]	IS-IS ルート テーブルを表示します。
show isis [*instance-tag*] **rrm** [*interface*] [**vrf** *vrf-name*]	IS-IS インターフェイスの再送信情報を表示します。
show isis [*instance-tag*] **srm** [*interface*] [**vrf** *vrf-name*]	IS-IS インターフェイスのフラッディング情報を表示します。
show isis [*instance-tag*] **ssn** [*interface*] [**vrf** *vrf-name*]	IS-IS インターフェイスの PSNP 情報を表示します。
show isis [*instance-tag*] {**ip** | **ipv6**} **summary-address**] [*ip-address*] | [*ip-prefix*] [**vrf** *vrf-name*]	IS-IS のサマリー アドレス情報を表示します。
show running-configuration isis	現在の実行中の IS-IS 設定を表示します。
show tech-support isis [**detail**]	IS-IS のテクニカル サポートの詳細情報を表示します。

IS-IS の監視

IS-IS の統計情報を表示するには、次のコマンドを使用します。

コマンド	目的
show isis [*instance-tag*] **adjacency** [*interface*] [**system-ID**] [**detail**] [**summary**] [**vrf** *vrf-name*]	IS-IS 隣接関係の統計情報を表示します。
show isis [*instance-tag*] **database** [**level-1** | **level-2**] [**detail**] | **summary**] [*lsip*] {[**adjacency** *id* {**ip** | **ipv6**} **prefix** *prefix*] [**router-id** *id*] [*zero-sequence*]} [**vrf** *vrf-name*]	IS-IS データベースの統計情報を表示します。
show isis [*instance-tag*] **statistics** [*interface*] [**vrf** *vrf-name*]	IS-IS インターフェイスの統計情報を表示します。
show isis {**ip** | **ipv6**} **route-map statistics redistribute** {**bgp** *id* | **eigrp** *id* | **isis** *id* | **ospf** *id* | **rip** *id* | **static**} [**vrf** *vrf-name*]	IS-IS 再配布の統計情報を表示します。
show isis ip route-map statistics distribute {**level-1** | **level-2**} **into** {**level-1** | **level-2**} [**vrf** *vrf-name*]	レベル間で配布されたルートに関する、IS-IS 配布統計情報を表示します。
show isis [*instance-tag*] **spf-log** [**detail**] [**vrf** *vrf-name*]	IS-IS SPF 計算の統計情報を表示します。

コマンド	目的
show isis [*instance-tag*] **traffic** [*interface*] [**vrf** *vrf-name*]	IS-IS トラフィックの統計情報を表示します。

IS-IS 設定の統計情報を消去するには、次のいずれかのタスクを実行します。

コマンド	目的
clear isis [*instance-tag*] **adjacency** [* | [*interface*] [**system-id** *id*]] [**vrf** *vrf-name*]	IS-IS 隣接関係の統計情報を消去します。
clear isis {**ip** | **ipv6**} **route map statistics redistribute** {**bgp** *id* | **direct** | **eigrp** *id* | **isis** *id* | **ospf** *id* | **rip** *id* | **static**} [**vrf** *vrf-name*]	IS-IS 再配布の統計情報を消去します。
clear isis route-map statistics distribute {**level-1** | **level-2**} **into** {**level-1** | **level-2**} [**vrf** *vrf-name*]	レベル間で配布されたルートに関する、IS-IS 配布統計情報を消去します。
clear isis [*instance-tag*] **statistics** [* | *interface*] [**vrf** *vrf-name*]	IS-IS インターフェイスの統計情報を消去します。
clear isis [*instance-tag*] **traffic** [* | *interface*] [**vrf** *vrf-name*]	IS-IS トラフィックの統計情報を消去します。

IS-IS の設定例

IS-IS を設定する例を示します。

```
router isis Enterprise
 is-type level-1
 net 49.0001.0000.0000.0003.00
 graceful-restart
 address-family ipv4 unicast
  default-information originate

interface ethernet 2/1
 ip address 192.0.2.1/24
 isis circuit-type level-1
 ip router isis Enterprise
```

関連項目

ルートマップの詳細については、Route Policy Manager の設定（587 ページ）を参照してください。

関連項目

第 10 章

基本的 BGP の設定

この章では、Cisco NX-OS デバイス上でボーダー ゲートウェイ プロトコル（BGP）を設定する方法について説明します

この章は、次の項で構成されています。

- 基本的な BGP について （319 ページ）
- BGP の前提条件 （330 ページ）
- 基本 BGP に関する注意事項と制約事項 （330 ページ）
- デフォルト設定 （332 ページ）
- CLI コンフィギュレーション モード （333 ページ）
- 基本的 BGP の設定 （335 ページ）
- ベーシック BGP の設定の確認 （351 ページ）
- BGP 統計情報のモニタリング （354 ページ）
- ベーシック BGP の設定例 （354 ページ）
- 関連項目 （354 ページ）
- 次の作業 （354 ページ）
- その他の参考資料 （355 ページ）

基本的な BGP について

Cisco NX-OS は BGP バージョン 4 をサポートします。BGP v4 に組み込まれているマルチ プロトコル拡張機能を使用すると、IP マルチキャスト ルートおよび複数のレイヤ 3 プロトコル アドレス ファミリに関するルーティング情報を BGP に伝送させることができます。BGP では、他の BGP 対応デバイスとの間で TCP セッションを確立するための、信頼できるトランスポート プロトコルとして TCP を使用します。

BGP ではパスベクトル ルーティング アルゴリズムを使用して、BGP 対応ネットワーク デバイスまたは BGP スピーカ間でルーティング情報を交換します。各 BGP スピーカはこの情報を使用して、特定の宛先までのパスを判別し、なおかつルーティング ループを伴うパスを検出して回避します。ルーティング情報には、宛先の実際のルート プレフィックス、宛先に対する自律システムのパス、およびその他のパス属性が含まれます。

BGP はデフォルトで、宛先ホストまたはネットワークへのベスト パスとして、1 つだけパスを選択します。各パスは、BGP ベストパス分析で使用される well-known mandatory、well-known discretionary、optional transitive の各属性を伝送します。BGP ポリシーを設定し、これらの属性の一部を変更することによって、BGP パス選択を制御できます。詳細については、ルート ポリシーおよび BGP セッションのリセット（359 ページ）を参照してください。

BGP は、ロード バランシングまたは等コスト マルチパス（ECMP）もサポートします。詳細については、「ロード シェアリングとマルチパス」の項を参照してください。

BGP 自律システム

自律システム（AS）とは、単一の管理エンティティにより制御されるネットワークです。自律システムは 1 つまたは複数の IGP および整合性のある一連のルーティング ポリシーを使用して、ルーティング ドメインを形成します。BGP は 16 ビットおよび 32 ビットの自律システム番号をサポートします。詳細については、「自律システム」を参照してください。

個々の BGP 自律システムは外部 BGP（eBGP）ピアリング セッションを通じて、ルーティング情報をダイナミックに交換します。同じ自律システム内の BGP スピーカは、内部 BGP（iBGP）を通じて、ルーティング情報を交換できます。

4 バイトの AS 番号のサポート

BGP は、プレーン テキスト表記法または AS ドット付き表記法の 2 バイトの自律システム（AS）番号、もしくはプレーン テキスト表記法の 4 バイトの AS 番号をサポートします。

4 バイトの AS 番号を使用して BGP が設定されている場合は、**route-target auto** VXLAN コマンドを使用できません。これは、AS 番号とともに（すでに 3 バイト値である）VNI がルート ターゲットの生成に使用されるためです。詳細については、『Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide』を参照してください。

アドミニストレーティブ ディスタンス

アドミニストレーティブ ディスタンスは、ルーティング情報源の信頼性を示す評価基準です。デフォルトで、BGP は表に示されたアドミニストレーティブ ディスタンスを使用します。

表 *22*：デフォルトの *BGP* アドミニストレーティブ ディスタンス

ディスタンス	デフォルト値	機能
外部	20	eBGP から学習したルートに適用されます。
内部	200	iBGP から学習したルートに適用されます。
ローカル	220	ルータを起点とするルートに適用されます。

（注） アドミニストレーティブディスタンスがBGPパス選択アルゴリズムに影響を与えることはありませんが、BGP で学習されたルートが IP ルーティング テーブルに組み込まれるかどうかを左右します。

詳細については、「アドミニストレーティブ ディスタンス」のセクションを参照してください。

BGP ピア

BGP スピーカーは他の BGP スピーカーを自動的に検出しません。ユーザ側で BGP スピーカ間の関係を設定する必要があります。BGP ピアは、別の BGP スピーカへのアクティブな TCP 接続を持つ BGP スピーカです。

BGP セッション

BGP は TCP ポート 179 を使用して、ピアとの TCP セッションを作成します。ピア間で TCP 接続が確立されると、各 BGP ピアは最初に相手と、それぞれのすべてのルートを交換し、BGP ルーティングテーブルを完成させます。初期交換以後、BGP ピアはネットワーク トポロジが変化したとき、またはルーティング ポリシーが変更されたときに、差分アップデートだけを送信します。更新と更新の間の非アクティブ期間には、ピアは「キープアライブ」と呼ばれる特別なメッセージを交換します。ホールドタイムは、は、次のBGPアップデートまたはキープアライブ メッセージを受信するまでに経過することが許容される、最大時間限度です。

Cisco NX-OS は、次のピア設定オプションをサポートします。

- 個別の IPv4 または IPv6 アドレス：BGP は、リモート アドレスと AS 番号が一致する BGP スピーカとのセッションを確立します。
- 単一 AS 番号の IPv4 または IPv6 プレフィックス ピア：BGP は、プレフィックスおよび AS 番号が一致する BGP スピーカとのセッションを確立します。
- ダイナミック AS 番号プレフィックス ピア：BGP は、プレフィックスと、設定済み AS 番号のリストに載っている AS 番号と一致する BGP スピーカとのセッションを確立します。

プレフィックス ピアおよびインターフェイス ピアのダイナミック AS 番号

Cisco NX-OS では、BGP セッションを確立する AS 番号の範囲またはリストを受け入れます。たとえば IPv4 プレフィックス 192.0.2.0/8 および AS 番号 33、66、99 を使用するように BGP を設定する場合、BGP は 192.0.2.1 および AS 番号 66 を使用してセッションを確立しますが、192.0.2.2 および AS 番号 50 からのセッションは拒否します。

Cisco NX-OS リリース 9.3(6) 以降、ダイナミック AS 番号のサポートは、プレフィックス ピアに加えてインターフェイス ピアにも拡張されています。IPv4 および IPv6 アドレス ファミリ向け IPv6 リンク ローカル経由の BGP インターフェイス ピアリングの設定 （389 ページ）を参照してください。

Cisco NX-OS では、セッションが確立されるまで内部 BGP（iBGP）または外部 BGP（eBGP）セッションとして、プレフィックス ピアをダイナミック AS 番号と関連付けません。iBGP および eBGP の詳細については、高度な BGP の設定（357 ページ）を参照してください。

（注） ダイナミック AS 番号プレフィックス ピア設定は、BGP テンプレートから継承した個々の AS 番号の設定よりも優先します。詳細については、高度な BGP の設定（357 ページ）を参照してください。

BGP ルータ ID

ピア間で BGP セッションを確立するには、BGP セッションの確立時に、OPEN メッセージで BGP ピアに送信されるルータ ID を BGP に設定する必要があります。BGP ルータ ID は 32 ビット値であり、IPv4 アドレスで表すことがよくあります。ルータ ID はユーザ側で設定できます。ルータ ID はデフォルトで、Cisco NX-OS によってルータのループバック インターフェイスの IPv4 アドレスに設定されます。ルータ上でループバック インターフェイスが設定されていない場合は、ルータ上の物理インターフェイスに設定されている最大の IPv4 アドレスが BGP ルータ ID を表すものとして、ソフトウェアによって選択されます。BGP ルータ ID は、ネットワーク内の BGP ピアごとに一意である必要があります。

BGP にルータ ID が設定されていない場合、BGP ピアとのピアリング セッションを確立できません。

各ルーティング プロセスには、ルータ ID が関連付けられています。ルータ ID は、システムのあらゆるインターフェイスに設定できます。ルータ ID を構成しなかった場合、Cisco NX-OS は次の基準に基づいてルータ識別子を選択します。

- Cisco NX-OS は、他のあらゆるインターフェイスよりも loopback0 を優先します。loopback0 が存在しなかった場合、Cisco NX-OS は、他のあらゆるインターフェイス タイプよりも、最初のループバック インターフェイスを優先します。
- ループバック インターフェイスを構成しなかった場合、Cisco NX-OS はルータ識別子として構成ファイルの最初のインターフェイスを使用します。Cisco NX-OS がルータ識別子を選択した後にいずれかのループバック インターフェイスを設定した場合は、ループバック インターフェイスがルータ識別子となります。ループバック インターフェイスが loopback0 ではなく、loopback0 を IP アドレスで設定した場合は、ルータ ID が loopback0 の IP アドレスに変更されます。
- ルータ ID の元であるインターフェイスが変更されると、新しい IP アドレスがルータ ID となります。他のどのインターフェイスの IP アドレスが変更されても、ルータ ID はまったく変更されません。

BGP パスの選択

BGP は、プレフィックスごとの複数パスの送受信と、このパスのアドバタイジングをサポートします。追加 BGP パスの設定については、高度な BGP の設定（357 ページ）を参照してください。

所定のネットワークでパスが追加または削除されるたびに、ベストパス アルゴリズムが実行されます。ベストパス アルゴリズムは、ユーザが BGP 設定を変更した場合にも実行されます。BGP は所定のネットワークで使用できる一連の有効パスの中から、最適なパスを選択します。

Cisco NX-OS は次の手順で、BGP ベストパス アルゴリズムを実行します。

1. 2 つのパスを比較し、どちらが適切かを判別します（「ステップ 1 - 「パスの比較ペア」セクションを参照）。
2. すべてのパスを探索し、全体として最適なパスを選択するためにパスを比較する順序を決定します（ステップ 2 -「比較順序の決定」セクションを参照）。
3. 新しいベスト パスを使用するに足るだけの差が新旧のベスト パスにあるかどうかを判別します（ステップ 3 -「ベストパス変更抑制の決定」セクションを参照）。

（注） 重要なのは、パート 2 で決定される比較順序です。3 つのパス A、B、C があるとします。Cisco NX-OS が A と B を比較する場合、A を選択します。Cisco NX-OS が B と C を比較する場合、B を選択します。しかし、Cisco NX-OS がA と C を比較した場合、A を選択しません。これは一部の BGP メトリックが同じネイバー自律システムからのパスだけに適用され、すべてのパスにわたっては適用されないからです。

パス選択には、BGP AS パス属性が使用されます。AS パス属性には、アドバタイズされたパスでたどる自律システム番号（AS 番号）のリストが含まれます。BGP 自律システムを自律システムの集合または連合に細分化する場合は、AS パスにローカル定義の自律システムを指定した連合セグメントが含まれます。

（注） VXLAN の導入では、BGP パス選択プロセスが使用されます。このプロセスは、ローカルパスからリモートパスへの通常の選択とは異なります。EVPN アドレスファミリの場合、BGP は MAC モビリティ属性のシーケンス番号を比較し（存在する場合）、より高いシーケンス番号のパスを選択します。比較対象の両方のパスに属性があり、シーケンス番号が同じである場合、BGP はローカルで生成されたパスよりもリモートピアから学習したパスを優先します。詳細については、『Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide』を参照してください。

BGP パス選択：パスびペアの比較

BGP ベストパス アルゴリズムの最初のステップでは、より適切なパスを判別するために 2 つのパスを比較します。次に、Cisco NX-OS が 2 つのパスを比較して、より適切なパスを判別する基本的なステップについて説明します。

1. Cisco NX-OS は、比較のために有効なパスを選択します（たとえば、到達不能なネクストホップがあるパスは無効です）。
2. Cisco NX-OS は、重みが最大のパスを選択します。
3. Cisco NX-OS は、ローカル プリファレンスが最大のパスを選択します。

4. パスの一方がローカル起点の場合、Cisco NX-OS はそのパスを選択します。
5. Cisco NX-OS は、AS パスが短い方のパスを選択します。

（注） AS パス長を計算するときに、Cisco NX-OS は連合セグメントを無視し、AS セットを 1 として数えます。詳細については、「AS 連合」の項を参照してください。

6. Cisco NX-OS は、起点が低い方のパスを選択します。IGP は EGP よりも低いと見なされます。
7. Cisco NX-OS は、Multi-Exit 識別子（MED）が小さい方のパスを選択します。

   パスのピア自律システムに関係なく、ベストパス アルゴリズムの MED 比較が必ず実行されるように、Cisco NX-OS を設定することもできます。詳細については、「最適パス アルゴリズムの調整」を参照してください。この設定を行わなかった場合、MED 比較が実行されるかどうかは、次のように比較する 2 つのパスの AS パス属性によって決まります。

   パスのピア自律システムに関係なく、ベストパス アルゴリズムの MED 比較が必ず実行されるように、Cisco NX-OS を設定することもできます。この設定を行わなかった場合、Cisco NX-OS によって MED 比較が実行されるかどうかは、次のように比較する 2 つのパスの AS パス属性によって決まります。

   1. パスに AS パスまたは AS_SET から始まる AS パスがない場合、パスは内部であり、Cisco NX-OS は他の内部パスに対して MED を比較します。
   2. AS パスが AS_SEQUENCE から始まる場合、ピア自律システムがシーケンスで最初の AS 番号になり、Cisco NX-OS は同じピア自律システムを持つ他のパスに対して MED を比較します。
   3. AS-path パス に連合セグメントだけが含まれている場合、または連合セグメントで始まり、AS_SET が続いている場合、パスは内部であり、Cisco NX-OS は他の内部パスに対して MED を比較します。
   4. AS パスが連合セグメントで始まり、AS_SEQUENCE が続いている場合、ピア自律システムが AS_SEQUENCE で最初の AS 番号になり、Cisco NX-OS は同じピア自律システムを持つ他のパスに対して MED を比較します。

（注） Cisco NX-OS がパスの指定された MED 属性を受信しなかった場合、Cisco NX-OS は欠落 MED が使用可能な最大値になるように、ユーザがベストパス アルゴリズムを設定していない限り、MED を 0 と見なします。詳細については、「最適パス アルゴリズムの調整」を参照してください。

   5. 非決定性の MED 比較機能がイネーブルの場合、ベスト-パス アルゴリズムでは Cisco IOS スタイルの MED 比較が使用されます。

8. 一方のパスが内部ピアから、他方のパスが外部ピアからの場合、Cisco NX-OS は外部ピアからのパスを選択します。

9. ネクスト ホップ アドレスへの IGP メトリックが異なるパスの場合、Cisco NX-OS は IGP メトリックが小さい方のパスを選択します。

10. Cisco NX-OS は、最後に実行したベストパス アルゴリズムによって選択されたパスを使用します。

    ステップ 1 ～ 9 のすべてのパス パラメータが同じ場合、最適パス アルゴリズムを構成し、「ルータ ID の比較」を構成して、両方のパスが eBGP であるときに、ルータ ID の比較を適用できます。その他のすべての場合、ルータ ID の比較はデフォルトで実行されます。

    詳細については、「最適パス アルゴリズムの調整」を参照してください。パスに発信元属性が含まれている場合、Cisco NX-OS はその属性をルータ ID として使用して比較します。発信もと属性が含まれていない場合、Cisco NX-OS はパスを送信したピアのルータ ID を使用します。パス間でルータ ID が異なる場合、Cisco NX-OS はルータ ID が小さい方のパスを選択します。

（注） 属性の送信元をルータ ID として使用する場合は、2 つのパスに同じルータ ID を設定することができます。また、同じピア ルータとの 2 つの BGP セッションが可能です。したがって、同じルータ ID で 2 つのパスを受信できます。

11. Cisco NX-OS は、クラスタ長が短いほうのパスを選択します。クラスタ リスト属性の指定されたパスを受け取らなかった場合、クラスタ長は 0 です。

12. Cisco NX-OS は、IP アドレスが小さい方のピアから受信したパスを選択します。ローカル発生のパス（再配布のパスなど）は、ピア IP アドレスが 0 になります。

（注） ステップ 9 以降が同じパスは、マルチパスを設定している場合、マルチパスに使用できます。詳細については、「ロード シェアリングとマルチパス」の項を参照してください。

BGPパス選択：比較の順序の決定

BGP ベストパス アルゴリズム実装の 2 番目のステップでは、Cisco NX-OS がパスを比較する順序を決定します。

1. Cisco NX-OS は、パスをグループに分けます。各グループ内で、Cisco NX-OS はすべてのパス間で MED を比較します。Cisco NX-OSは、「ステップ 1：パス ペアの比較」と同じルールを使用して、2 つのパス間で MED を比較できるかどうかを判断します。この比較では通常、ネイバー自律システムごとに 1 つずつグループが選択されます。**bgp bestpath med always** コマンドを設定すると、Cisco NX-OS はすべてのパスが含まれた 1 グループだけを選択します。

2. Cisco NX-OS は、常に最適な方を維持しながら、グループのすべてのパスを反復することによって、各グループのベスト パスを決定します。Cisco NX-OS は、各パスをそれまでの一時的なベスト パスと比較します。それまでのベスト パスよりも適切な場合は、そのパスが新しく一時的なベスト パスになり、Cisco NX-OS はグループの次のパスと比較します。

3. Cisco NX-OS は、ステップ 2 の各グループで選択されたベスト パスからなる、パス セットを形成します。Cisco NX-OS は、このパス セットでもステップ 2 と同様にそれぞれの比較を繰り返すことによって、全体としてのベスト パスを選択します。

BGP パス選択：最適パス変更抑制の決定

実装の次のパートでは、Cisco NX-OS が新しい最適パスを使用するのか抑制するのかを決定します。新しいベストパスが古いパスとまったく同じ場合、ルータは引き続き既存のベストパスを使用できます（ルータ ID が同じ場合）。Cisco NX-OS では引き続き既存のベスト パスを使用することによって、ネットワークにおけるルート変更を回避できます。

抑制機能をオフにするには、ルータ ID を比較するようにベストパスアルゴリズムを設定します。詳細については、「最適パス アルゴリズムの調整」を参照してください。この機能を設定すると、新しいベスト パスが常にに既存のベスト パスよりも優先されます。

次の条件が発生した場合に、ベスト パス変更を抑制できません。

- 既存のベスト パスが無効になった。
- 既存または新しいベスト パスを内部（または連合）ピアから受信したか、またはローカルに発生した（再配布などによって）。
- 同じピアからパスを受信した（パスのルータ ID が同じ）。
- パス間で重み値、ローカルプリファレンス、オリジン、またはネクストホップアドレスに対する IGP メトリックが異なっている。
- パス間で MED が異なっている。

BGP およびユニキャスト RIB

BGP はユニキャスト RIB（ルーティング情報ベース）と通信して、ユニキャスト ルーティング テーブルに IPv4 ルートを格納します。ベスト パスの選択後、ベスト パスの変更をルーティング テーブルに反映させる必要があると BGP が判別した場合、BGP はユニキャスト RIB にルート アップデートを送信します。

BGP はユニキャスト RIB における BGP ルートの変更に関して、ルート通知を受け取ります。さらに、再配布をサポートする他のプロトコル ルートに関するルート通知を受け取ります。

BGP はネクストホップの変更に関する通知も、ユニキャスト RIB から受け取ります。BGP はこれらの通知を使用して、ネクストホップ アドレスへの到達可能性および IGP メトリックを追跡します。

ユニキャスト RIB でネクストホップ到達可能性または IGP メトリックが変更されるたびに、BGP は影響を受けるルートについて、ベスト パス再計算を開始させます。

BGP は IPv6 ユニキャスト RIB と通信し、IPv6 ルートについて、これらの動作を実行します。

BGP プレフィックス独立コンバージェンス

BGPプレフィックス独立コンバージェンス（PIC）エッジ機能は、リンク障害が発生した場合に、BGP バックアップ パスへの BGP IP ルートのコンバージェンスを高速化します。

BGP PIC エッジ機能により、ネットワーク障害後の BGP コンバージェンスが向上します。このコンバージェンスは、IP ネットワークのエッジ障害に適用されます。この機能は、ルーティング情報ベース（RIB）と転送情報ベース（FIB）にバックアップ パスを作成して保存します。これによって、プライマリ パスの障害が発生した場合に、ただちにバックアップ パスが引き継ぐことができ、フォワーディング プレーンの迅速なフェールオーバーが可能になります。BGP PIC エッジは、IPv4 アドレス ファミリのみをサポートします。

BGP PIC エッジが設定されている場合、BGP は、プライマリ ベスト パスに加えて、2 番目のベスト パス（バックアップ パス）も計算します。BGP は、PIC サポートを持つプレフィクスのベスト パスとバックアップ パスの両方を BGP RIB にインストールします。また BGP は、API を介してリモートの次のホップとともにバックアップ パスを URIB にダウンロードし、その後バックアップとしてマークされたネクスト ホップで FIB を更新します。バックアップ パスにより、単一のネットワーク障害に対処する高速再ルーティング機能が提供されます。

この機能は、ローカル インターフェイスとリモート インターフェイス/リンクの両方の障害を検出して、バックアップ パスが使用されるようにします。

BGP PIC エッジは、ユニパスとマルチパスの両方をサポートします。

BGP PIC エッジ ユニパス

次の図に、BGP PIC エッジ ユニパスのトポロジを示します。

図 *27 : BGP PIC* エッジ ユニパス

この図では次のようになっています。

- S2-S4とS3-S5の間はeBGPセッションです。
- S2-S3 の間は iBGP セッションです。
- S1からのトラフィックはS2を使用し、またe1インターフェイスを使用してZ1..Znプレフィックスに到達します。
- S2には、Z1…Znに到達するための 2 つのパスがあります。
  - S4 を経由するプライマリ パス
  - S5 を経由するバックアップ パス

この例では、S3 が S2 に対し、到達すべきプレフィクス Z1…Zn をアドバタイズします（それ自身をネクスト ホップとして）。BGP PIC エッジが有効になっている場合、S2 の BGP は、AS6500へのベストパス（S4 経由）とバックアップパス（S3 または S5 を経由）の両方を RIB にインストールします。その後、RIBは両方のルートを FIBにダウンロードします。

S2-S4 のリンクがダウンすると、S2 上の FIB がリンク障害を検出します。その場合、自動的にプライマリパスからバックアップに切り替えられ、新しいネクストホップ S3 がポイントされます。トラフィックは、FIB 内のローカルの高速再コンバージェンスにより迅速に再ルーティングされます。リンク障害イベントを学習した後、S2 上の BGP はベストパス（以前のバックアップ パス）を再計算し、RIB からネクスト ホップ S4 を削除し、S3 をプライマリ ネクスト ホップとして RIB に再インストールします。また、新しいバックアップあればそれも計算し、RIB に通知します。BGP PIC エッジ機能のサポートにより、FIB はプライマリ ルートでのリンク障害の検出時に、BGP が新しいベストパスを選択してコンバージェンスするまで待機することなく、使用可能なバックアップ ルートに瞬時に切り替えます。こうして、高速な再ルーティングを実現しています。

マルチパスを持つ BGP PIC エッジ

次の図に、BGP PIC エッジ マルチパス トポロジを示します。

図 *28 : BGP PIC* エッジ マルチパス

上記のトポロジでは、次のように所定のプレフィックスに 6 つのパスがあります。

- eBGP パス：e1、e2、e3
- iBGPパス：i1、i2、i3

優先順位は、e1 > e2 > e3 > i1 > i2 > i3 です。

考えられるマルチパスの状況は次のとおりです。

- 設定されたマルチパスなし：
  - ベストパス = e1
  - マルチパス-セット = []
  - バックアップ パス = e2
  - PIC 挙動：e1 が失敗すると、e2 がアクティブになります。
- 双方向の eBGP マルチパスが設定されている
  - ベストパス = e1
  - マルチパス-セット = [e1, e2]
  - バックアップ パス = e3
  - PIC 挙動：アクティブなマルチパスが相互にバックアップされます。すべてのマルチパスが失敗すると、e3 がアクティブになります。
- 3 方向の eBGP マルチパスが設定されている
  - ベストパス = e1

- マルチパス-セット = [e1, e2, e3]
- バックアップ パス = i1
- PIC 挙動：アクティブなマルチパスが相互にバックアップされます。すべてのマルチパスが失敗すると、i1 がアクティブになります。

- 4 方向の eiBGP マルチパスが設定されている
  - – ベストパス = e1
  - – マルチパスセット = [e1, e2, e3, i1]
  - – バックアップパス = i2
  - –PIC 挙動：アクティブなマルチパスが相互にバックアップされます。すべてのマルチパスが失敗すると、i2 がアクティブになります。

等コスト マルチパス（ECMP）がイネーブルになっている場合、どのマルチパスもバックアップ パスとして選択されません。

バックアップ パスを使用するマルチパスのシナリオでは、すべてのアクティブなマルチパスで同時障害が発生しても、高速コンバージェンスは生じません。

BGP PIC コア

コアの BGP Prefix Independent Convergence（PIC）は、ネットワーク障害後の BGP コンバージェンスを向上させます。たとえば、プロバイダーエッジ（PE）でリンクに障害が発生した場合、ルーティング情報ベース（RIB）は新しいネクスト ホップで転送情報ベース（FIB）を更新します。FIB は、失敗したネクスト ホップを指しているすべての BGP プレフィックス、新しいネクスト ホップを指すように更新する必要があります。これは、時間とリソースを消費する可能性があります。BGP PIC コアを有効にすると、FIB 内でプレフィックスが階層的にプログラムされます。すべてのプレフィックスは、再帰ネクスト ホップではなく、ECMP グループを指します。同じ障害が発生した場合、FIB は、プレフィックスを更新せず、新しいネクスト ホップを指すよう ECMP グループを更新するだけで済みます。これにより、BGP は IGP コンバージェンスを即座に活用できます。

BGP PIC の機能サポート マトリクス

表 *23 : BGP PIC* の機能サポート マトリクス

BGP PIC	IPv4 ユニキャスト	IPv6 ユニキャスト
エッジ ユニパス	はい	非対応
マルチパスを持つエッジ（複数のアクティブ ECMP、バックアップ 1 つのみ）	はい	非対応
コア	はい	はい

BGP の仮想化

BGP は、仮想ルーティングおよび転送（VRF）インスタンスをサポートします。

BGP の前提条件

BGP を使用するには、次の前提条件を満たしている必要があります。

- BGP を有効にする必要があります（「BGP の有効化」の項を参照）。
- システムに有効なルータ ID を設定しておく必要があります。
- Regional Internet Registry（RIR）によって割り当てられたか、またはローカル管理の AS 番号を取得しておく必要があります。
- 再帰ネクストホップ解決に対応できる IGP を 1 つ以上設定する必要があります。
- BGP セッションを確立するネイバー環境で、アドレス ファミリを設定する必要があります。

基本 BGP に関する注意事項と制約事項

BGP 設定時の注意事項および制約事項は、次のとおりです。

- 十分な規模（ピアあたり数百のピアや数千のルートなど）では、デフォルトの 5分 間の古いパス タイマーでは、BGP コンバージェンスが完了しないためにタイマーが期限切れになる可能性があるため、グレースフル リスタート メカニズムが失敗する可能性があります。次のコマンドを使用して、コンバージェンス プロセスにかかる実際の時間を確認します。

```
switch# show bgp vrf all all neighbors | in First|RIB
  Last End-of-RIB received 0.022810 after session start
  Last End-of-RIB sent 00:08:36 after session start
  First convergence 00:08:36 after session start with 398002 routes sent
```

- Cisco NX-OS 9.3(5) 以降では、vPC ピアへの TTL 値が 1 のパケットがハードウェア転送されます。
- レコード オプション（-Cr）を指定して SNMP バルクウォークを使用する場合、大規模なルーティング テーブル（250 K以上）では、SNMP パフォーマンスの低下を避けるために 10 個を超えるレコードを使用しないでください。
- プレフィックス リスト内の名前は、大文字と小文字が区別されません。一意の名前を使用することを推奨します。大文字と小文字を変更して同じ名前を使用しないでください。たとえば、CTCPrimaryNetworks と CtcPrimaryNetworks は 2 つの異なるエントリではありません。
- サポートされるプラットフォームに関する詳細は、サポートされるプラットフォーム （3 ページ） を参照してください。
- ダイナミック AS 番号プレフィックス ピア設定は、BGP テンプレートから継承した個々の AS 番号の設定よりも優先します。

- AS 連合でプレフィックス ピアにダイナミック AS 番号を設定した場合、BGP はローカル連合の AS 番号のみでセッションを確立します。
- ダイナミック AS 番号プレフィックス ピアで作成された BGP セッションは、設定済みの eBGP マルチホップ存続可能時間（TTL）値や直接接続ピアに対するディセーブル済みのチェックを無視します。
- ルータ ID の自動変更およびセッション フラップを避けるために、BGP 用のルータ ID を設定します。
- ピアごとに最大プレフィックス設定オプションを使用し、受信するルート数および使用するシステム リソース数を制限してください。
- update-source を設定し、BGP/eBGP マルチホップ セッションでセッションを確立します。
- 再配布を設定する場合は、BGP ポリシーを指定します。
- VRF 内で BGP ルータ ID を定義します。
- IPv6ネイバーの場合は、VRFごとにルータIDを設定することを推奨します。VRFにIPv4インターフェイスがない場合、IPv6 BGPネイバーはルータIDがIPv4アドレスである必要があるため、アップしません。数値が最小のループバックIPv4アドレスがルータIDとして選択されます。ループバックアドレスが存在しない場合は、VRFインターフェイスから最も小さいIPアドレスが選択されます。これが存在しない場合、BGPネイバー関係は確立されません。
- キープアライブおよびホールドタイマーの値を小さくすると、BGPセッションフラップが発生する可能性があります。
- **advertisement-interval**コマンドを使用すると、BGP ルーティング アップデートを送信する最小ルート アドバタイズメント インターバル（MRAI）を設定できます。
- **show ip bgp** コマンドは BGP 設定の確認に使用できますが、代わりに **show bgp** コマンドを使用することを推奨します。
- ルートマップ削除機能は、BGPに関連付けられたルートマップ全体の削除をブロックするメカニズムを追加します。ルート マップの削除がブロックされても、ルート マップ ステートメントへの変更は引き続き許可されます。
- ルート マップに複数のシーケンスがある場合、少なくとも 1 つのシーケンスが使用可能になるまで、ユーザーはルート マップ シーケンスを削除できます。
- ユーザーは、クライアントからのルート マップの前方参照ケースを持つことができます。ただし、ルートマップが作成されて関連付けられると、ルートマップの削除はブロックされます。
- ブロック削除機能は、ノブを使用して動的に構成できます。
- ルート マップへの BGP アソシエーションを削除すること、および単一のトランザクション ペイロードでルート マップ自体を削除することは許可されています。
- ルート マップに BGP アソシエーションを追加することが許可されており、ルート マップの削除に対してエラーをスローする必要があります。

- 以下は、デュアル ステージに関連する動作のリストです。
  - ノブと削除が同時に発生した場合、デュアル ステージは検証し、コミットせずにエラーをスローする必要があります。
  - ノブはすでに存在し、ルートマップ削除がデュアル ステージで発生する場合、エラーをスローする必要があります。
  - ルートマップと CLI ノブが異なる順序のシングル コミットである場合、エラーをスローする必要があります。
  - ノブが有効になっておらず、ルート マップの削除がデュアル ステージで発生した場合は、正常に実行する必要があります。
  - 1回のベリファイで、「cliノブが無効かつルートマップの削除」が実行された場合、ルートマップの削除が許可されます。
- BGP テンプレートで使用されるルート マップがいずれの BGP ネイバーにも継承されない場合、ルート マップ全体の削除は引き続きブロックされます。
- Cloudscale IPv6 リンクローカル BGP のサポートには、512 を超える ing-sup TCAM リージョンを切り分ける必要があります (これを有効にするには、リロードが必要です)。
- Cisco NX-OS リリース 10.3(1)F 以降、BGP は Cisco Nexus 9808 プラットフォーム スイッチでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、BGP は Cisco Nexus 9804 プラットフォーム スイッチでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、BGP は、Cisco Nexus 9808 および 9804 スイッチを搭載した Cisco Nexus X98900CD-A および N9KX9836DM-A ライン カードでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降では、BGP ルートのカスタム分離モードで route-map を構成できます。

デフォルト設定

表 *24* : デフォルトの *BGP* パラメータ

パラメータ	デフォルト
BGP 機能	ディセーブル
キープアライブインターバル	60 秒
ホールド タイマー	180 秒
BGP PIC エッジ	ディセーブル
Auto-summary	常に無効

パラメータ	デフォルト
同期	常に無効

CLI コンフィギュレーション モード

以下の項では、BGP に対応する各 CLI コンフィギュレーション モードの開始方法について説明します。現行のモードで **?** コマンドを入力すると、そのモードで使用可能なコマンドを表示できます。

グローバル コンフィギュレーション モード

グローバル コンフィギュレーション モードは、BGP プロセスを作成したり、AS 連合、ルート ダンプニングなどの拡張機能を設定したりする場合に使用します。詳細については、高度な BGP の設定 （357 ページ）を参照してください。

次に、ルータ コンフィギュレーション モードを開始する例を示します。

```
switch# configuration
switch(config)# router bgp 64496
switch(config-router)#
```

BGP は VRF をサポートしています。ネットワークで VRF を使用する場合は、適切な VRF 内で BGP を設定できます。設定の詳細については、「仮想化の設定」の項を参照してください。

次に、VRF コンフィギュレーション モードを開始する例を示します。

```
switch(config)# router bgp 64497
switch(config-router)# vrf vrf_A
switch(config-router-vrf)#
```

アドレス ファミリ設定モード

任意で、BGP がサポートするアドレス ファミリを設定できます。アドレス ファミリ用の機能を設定する場合は、ルータ 設定モードで address-family コマンドを使用します。ネイバーに対応する特定のアドレス ファミリを設定する場合は、ネイバー設定モードで address-family コマンドを使用します。

ルート再配布、アドレス集約、ロード バランシングなどの拡張機能を使用する場合は、アドレス ファミリを設定する必要があります。

次に、ルータ設定モードからアドレス ファミリ設定モードを開始する例を示します。

```
switch(config)# router bgp 64496
switch(config-router)# address-family ipv6 unicast
switch(config-router-af)#
```

次に、VRF を使用している場合に、VRF アドレス ファミリ設定モードを開始する例を示します。

```
switch(config)# router bgp 64497
switch(config-router)# vrf vrf_A
```

```
switch(config-router-vrf)# address-family ipv6 unicast
switch(config-router-vrf-af)#
```

ネイバー コンフィギュレーション モード

Cisco NX-OS には、BGP ピアを設定するためのネイバー コンフィギュレーション モードがあります。ネイバーコンフィギュレーションモードを使用して、ピアのあらゆるパラメータを設定できます。

次に、ネイバー コンフィギュレーション モードを開始する例を示します。

```
switch(config)# router bgp 64496
switch(config-router)# neighbor 192.0.2.1
switch(config-router-neighbor)#
```

次に、VRF ネイバー コンフィギュレーション モードを開始する例を示します。

```
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)#
```

ネイバー アドレス ファミリ コンフィギュレーション モード

アドレスファミリ固有のネイバー設定を入力し、ネイバーのアドレスファミリをイネーブルにするには、ネイバー コンフィギュレーション サブモード内のアドレス ファミリ コンフィギュレーション サブモードを使用できます。このモードは、所定のネイバーに認められるプレフィックス数の制限、eBGP のプライベート AS 番号の削除といった拡張機能に使用します。

RFC 5549 が導入されているため、IPv6 アドレスを持つネイバーに IPv4 アドレス ファミリを設定できます。

この例は、IPv4 アドレスでネイバーのための IPv4 ネイバー アドレス ファミリ設定モードを入力する方法を示します。

```
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)#
```

この例は、IPv6 アドレスでネイバーのための IPv4 ネイバー アドレス ファミリ設定モードを入力する方法を示します。

```
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)#
```

この例は、IPv4 アドレスでネイバーのための VRF IPv4 ネイバー アドレス ファミリ設定モードを入力する方法を示します。

```
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)#
```

この例は、IPv6 アドレスでネイバーのための VRF IPv4 ネイバー アドレス ファミリ設定モードを入力する方法を示します。

```
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)#
```

基本的 BGP の設定

ベーシック BGP を設定するには、BGP を有効にして、BGP ピアを設定する必要があります。ベーシック BGP ネットワークの設定は、いくつかの必須作業と多数の任意の作業からなります。BGP ルーティング プロセスおよび BGP ピアの設定は必須です。

（注） Cisco IOS の CLI に慣れている場合、この機能の Cisco NX-OS コマンドは従来の Cisco IOS コマンドと異なる点があるため注意が必要です。

BGPの有効化

BGP を設定するには、その前に BGP を有効にする必要があります。

手順の概要

1. **configure terminal**
2. **[no] feature bgp**
3. （任意） **show feature**
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例:
	`switch# configure terminal`
	`switch(config)#`
		設定モードに入ります。

Step 2	**[no] feature bgp** 例: `switch(config)# feature bgp`	BGP を有効にします。 この機能を無効化するには、このコマンドの **no** 形式を使用します。

	コマンドまたはアクション	目的
Step 3	（任意） **show feature** 例： `switch(config)# show feature`	有効および無効にされた機能を表示します。
Step 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

BGP インスタンスの作成

BGP インスタンスを作成し、BGP インスタンスにルータ ID を割り当てることができます。詳細については、「BGP ルータ ID」の項を参照してください。

始める前に

- BGP をイネーブルにする必要があります（「BGP のイネーブル化」の項を参照）。
- BGP はルータ ID（設定済みループバック アドレスなど）を取得できなければなりません。

手順の概要

1. **configure terminal**
2. [**no**] **router bgp** {*autonomous-system-number* | *auto*}
3. **router-id** {*ip-address* | *auto*}
4. （任意） **address-family** {**ipv4**|**ipv6**} {**unicast**|**multicast**}
5. （任意） **network** {*ip-address/length* | *ip-address* **mask** *mask*} [**route-map** *map-name*]
6. （任意） **show bgp all**
7. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

Step 2	[**no**] **router bgp** {*autonomous-system-number* | *auto*} 例：	BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または

	コマンドまたはアクション	目的
	`switch(config)# router bgp 64496` `switch(config-router)#`	32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。 auto オプションは、システム MAC アドレスに基づいて 4 バイトのプライベート自律システム番号を自動的に生成します。 BGP プロセスおよび関連する設定を削除するには、このコマンドで no オプションを使用します。
Step 3	**router-id** {*ip-address* | *auto*} 例： `switch(config-router)# router-id 192.0.2.255`	（任意）BGP ルータ ID を設定します。この IP アドレスによって、この BGP スピーカを特定します。 「auto」オプションは、システム MAC アドレスに基づく BGP ルータ ID を有効にします。

Step 4	（任意） **address-family** {**ipv4**|**ipv6**} {**unicast**|**multicast**}
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		IPv4 または IPv6 アドレス ファミリに対してグローバルアドレス ファミリ コンフィギュレーション モードを開始します。

Step 5	（任意） **network** {*ip-address/length* | *ip-address* **mask** *mask*} [**route-map** *map-name*]
	例：
	`switch(config-router-af)# network 10.10.10.0/24`
	例：
	`switch(config-router-af)# network 10.10.10.0 mask 255.255.255.0`
		ネットワークを、この自律システムに対してローカルに設定し、BGP ルーティング テーブルに追加します。
		エクステリア プロトコルの場合、network コマンドでアドバタイズするネットワークを制御します。内部プロトコルは **network** コマンドを使用して、アップデートの送信先を決定します。

Step 6	（任意） **show bgp all** 例： `switch(config-router-af)# show bgp all`	すべての BGP アドレス ファミリに関する情報を表示します。
Step 7	（任意） **copy running-config startup-config** 例： `switch(config-router-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、IPv4 ユニキャスト アドレス ファミリを指定して BGP をイネーブルに設定し、アドバタイズするネットワークを 1 つ追加する例を示します。

```
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
```

BGP インスタンスの再起動

BGPインスタンスを再起動し、そのインスタンスのすべてのピアセッションをクリアできます。

BGP インスタンスを再起動し、関連付けられたすべてのピアを削除するには、次のコマンドを使用します。

手順の概要

1. **restart bgp***instance-tag*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**restart bgp***instance-tag* 例： `switch(config)# restart bgp 201`	BGP インスタンスを再起動し、すべてのピアリングセッションをリセットまたは再確立します。

BGP のシャットダウン

設定を維持しながら、BGP プロトコルをシャット ダウンして BGP を正常に無効にできます。

BGP をシャットダウンするには、ルータ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **shutdown**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**shutdown** 例： `switch(config-router)# shutdown`	BGP インスタンスを再起動し、すべてのピアリングセッションをリセットまたは再確立します。

BGP ピア設定

BGP プロセス内で BGP ピアを設定できます。BGP ピアごとに、関連付けられたキープアライブタイマーとホールドタイマーがあります。これらのタイマーは、グローバルに設定することも、BGP ピアごとに設定することもできます。ピア設定はグローバル設定を上書きします。

（注） ピアごとに、ネイバー コンフィギュレーション モードでアドレス ファミリを設定する必要があります。

始める前に

- BGP を有効にする必要があります（「BGP の有効化」の項を参照）。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor** {*ip-address* | *ipv6-address*} **remote-as** {*as-number* | *external* | *internal*}
4. **remote-as** {*as-number* | *external* | *internal*}
5. （任意） **description** *text*
6. （任意） **timers***keepalive-time hold-time*
7. （任意） **shutdown**
8. **address-family**{**ipv4**|**ipv6**} {**unicast**|**multicast**}
9. （任意） **weight** *value*
10. （任意） **show bgp** {**ipv4**|**ipv6**} {**unicast**|**multicast**} **neighbors**
11. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

Step 2	**router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

	コマンドまたはアクション	目的

Step 3	**neighbor** {*ip-address* | *ipv6-address*} **remote-as** {*as-number* | *external* | *internal*}
	例：
	`switch(config-router)# neighbor 209.165.201.1 remote-as 64497`
	`switch(config-router)# neighbor`
		リモート BGP ピアの IPv4 アドレスまたは IPv6 アドレスおよび AS 番号を設定します。*The ip-address* 形式は x.x.x.x です。*ipv6-address* の形式は A:B::C:D です。
		remote-as 値を手動で指定することなく、external および internal オプションを使用すると、eBGP および iBGP セッションを確立できます。

Step 4	**remote-as** {*as-number* | *external* | *internal*} 例： `switch(config-router-neighbor)# remote-as 64497`	リモート外部 BGP ピアの AS 番号を構成します。 remote-as 値を手動で指定することなく、external および internal オプションを使用すると、eBGP および iBGP セッションを確立できます。

Step 5	（任意） **description** *text*
	例：
	`switch(config-router-neighbor)# description Peer Router B`
	`switch(config-router-neighbor)#`
		ネイバーの説明を追加します。最大 80 文字の英数字ストリングを使用できます。

Step 6	（任意） **timers***keepalive-time hold-time* 例： `switch(config-router-neighbor)# timers 30 90`	ネイバーのキープアライブおよびホールドタイムを表す BGP タイマー値を追加します。指定できる範囲は 0 ～ 3600 秒です。デフォルト値は、キープアライブ タイムで 60 秒、ホールド タイムで 180 秒です。 （注） ホールドタイマーが 10 秒以下の BGP セッションは、BGP セッションが 60 秒以上稼働するまで有効になりません。セッションが 60 秒間稼働すると、ホールドタイマーは構成どおりに動作します。
Step 7	（任意） **shutdown** 例： `switch(config-router-neighbor)# shutdown`	この BGP ネイバーを管理目的でシャットダウンします。このコマンドによって、BGP ネイバー セッションの自動通知およびセッション リセットが開始されます。

Step 8	**address-family**{**ipv4**|**ipv6**} {**unicast**|**multicast**}
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		ユニキャスト IPv4 または IPv6 アドレス ファミリに対応するネイバー アドレス ファミリ コンフィギュレーション モードを開始します。

Step 9	（任意） **weight** *value* 例：	このネイバーからのルートのデフォルトの重みを設定します。範囲は 0 ～ 65535 です。

	コマンドまたはアクション	目的
	`switch(config-router-neighbor-af)# weight 100`	このネイバーから学習したすべてのルートに、まず重みが割り当てられます。特定のネットワークへのルートが複数ある場合、最大の重みを持つルートが優先ルートとして選ばれます。**set weight route-map** コマンドで割り当てられた重みは、このコマンドで割り当てられた重みを上書きします。 BGP ピア ポリシー テンプレートを指定した場合、テンプレートのメンバーすべてが、このコマンドで設定された特性を継承します。
Step 10	（任意） **show bgp** {**ipv4**|**ipv6**} {**unicast**|**multicast**} **neighbors** 例： `switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors`	BGP ピアに関する情報を表示します。
Step 11	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、BGP ピアの設定例を示します。

```
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)# weight 100
switch(config-router-neighbor-af)# copy running-config startup-config
```

プレフィックス ピアのダイナミック AS 番号の設定

BGP プロセス内で複数の BGP ピアを設定できます。BGP セッションの確立をルート マップの単一の AS 番号または複数の AS 番号に制限できます。

プレフィックス ピアのダイナミック AS 番号を介して設定された BGP セッションは、**ebgp-multihop** を無視します コマンドと **disable-connected-check** コマンドを使用する必要があります。

ルート マップの AS 番号のリストは変更できますが、ルート マップ名を変更するには no neighbor コマンドを使用する必要があります。設定されたルート マップの AS 番号に変更を加えた場合、新しいセッションのみに影響します。

始める前に

- BGP を有効にする必要があります（「BGP の有効化」の項を参照）。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor** *prefix* **remote-as route-map** *map-name*
4. **neighbor-as** *as-number*
5. （任意） **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors**
6. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

Step 2	**router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

Step 3	**neighbor** *prefix* **remote-as route-map** *map-name*
	例：
	`switch(config-router)# neighbor 192.0.2.0/8 remote-as routemap BGPPeers`
	`switch(config-router-neighbor)#`
		IPv4 プレフィックスまたは IPv6 プレフィックス、およびリモート BGP ピアの受け付けられた AS 番号のリストのルート マップを設定します。IPv4 の *prefix* 形式は、x.x.x.x/長さ長さの範囲は 1 ～ 32 です。IPv6 の場合、*prefix* の形式は「A:B::C:D/長さ」です。長さの範囲は 1 ～ 128 です。
		マップ-名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。

Step 4	**neighbor-as** *as-number* 例： `switch(config-router-neighbor)# remote-as 64497`	リモート BGP ピアの AS 番号を設定します。
Step 5	（任意） **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors** 例：	BGP ピアに関する情報を表示します。

	コマンドまたはアクション	目的
	`switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors`	
Step 6	（任意） **copy running-config startup-config** 例: `switch(config-router-neighbor-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、プレフィックス ピアのダイナミック AS 番号を設定する例を示します。

```
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-af)# end
switch# copy running-config startup-config
```

ルート マップについては、Route Policy Manager の設定 （587 ページ）を参照してください。

BGP PIC エッジの設定

BGP PIC エッジを設定するには、次の手順に従います。

（注） BGP PIC エッジ機能は、IPv4 アドレス ファミリのみをサポートします。

始める前に

BGP をイネーブルにする必要があります（「BGP のイネーブル化」の項を参照）。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **address-family ipv4 unicast**
4. **[no] additional-paths install backup**
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

Step 2	**router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

Step 3	**address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		IPv4 アドレス ファミリに対応するアドレス ファミリ構成モードを開始します。

Step 4	**[no] additional-paths install backup**
	例：
	`switch(config-router-af)#`
	`[no] additional-paths install backup`
		ルーティング テーブルにバックアップ パスをインストールする BGP をイネーブルにします。

Step 5	（任意） **copy running-config startup-config**
	例：
	`switch(config-router-af)# end`
	`switch# copy running-config startup-config`
		この設定変更を保存します。

例

次の例は、IPv4 ネットワークで BGP PIC エッジをサポートするように、デバイスを設定する方法を示しています。

```
interface Ethernet2/2
 ip address 1.1.1.5/24
 no shutdown

interface Ethernet2/3
 ip address 2.2.2.5/24
 no shutdown

router bgp 100
 address-family ipv4 unicast
  additional-paths install backup
```

```
 neighbor 2.2.2.6
   remote-as 100
   address-family ipv4 unicast
```

BGP が 2 つのネイバー（1.1.1.6 と 2.2.2.6）から同じプレフィックス（99.0.0.0/24 など）を受信した場合、両方のパスが URIB にインストールされます。一方はプライマリ パスになり、もう一方はバックアップ パスになります。

BGP 出力：

```
switch(config)# show ip bgp 99.0.0.0/24
BGP routing table information for VRF default, address family IPv4 Unicast BGP routing table
 entry
for 99.0.0.0/24, version 4
Paths: (2 available, best #2)
Flags: (0x00001a) on xmit-list, is in urib, is best urib route

Path type: internal, path is valid, not best reason: Internal path, backup path AS-Path:
200 , path
sourced external to AS
2.2.2.6 (metric 0) from 2.2.2.6 (2.2.2.6)
Origin IGP, MED not set, localpref 100, weight 0

Advertised path-id 1
Path type: external, path is valid, is best path AS-Path: 200 , path sourced external to
AS
1.1.1.6 (metric 0) from 1.1.1.6 (99.0.0.1)
Origin IGP, MED not set, localpref 100, weight 0

Path-id 1 advertised to peers: 2.2.2.6
```

URIB 出力：

```
switch(config)# show ip route 99.0.0.0/24
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>
99.0.0.0/24, ubest/mbest: 1/0
*via 1.1.1.6, [20/0], 14:34:51, bgp-100, external, tag 200
via 2.2.2.6, [200/0], 14:34:51, bgp-100, internal, tag 200 (backup)
```

UFIB 出力：

```
switch# show forwarding route 123.1.1.0 detail module 8
Prefix 123.1.1.0/24, No of paths: 1, Update time: Wed Jul 11 19:00:12 2018
Vobj id: 141   orig_as: 65002   peer_as: 65100  rnh: 10.3.0.2
10.4.0.2           Ethernet8/4         DMAC: 0018.bad8.4dfd
packets: 2  bytes: 3484   Repair path  10.3.0.2   Ethernet8/3 DMAC: 0018.bad8.4dfd packets:
 0
bytes: 1
```

BGP PIC コアの設定

BGP PIC Core を設定するには、次のステップに従います。

手順の概要

1. **configure terminal**
2. **[no] system pic-core**
3. **copy running-config startup-config**
4. **reload**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例： `switch# configure terminal`	グローバルコンフィギュレーションモードを開始します。
Step 2	**[no] system pic-core** 例： `switch(config)# system pic-core`	PIC の有効化を管理します。
Step 3	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。
Step 4	**reload** 例： `switch(config)# reload`	デバイス全体をリブートします。

BGP 情報の消去

BGP 情報を消去するには、次のコマンドを使用します。

コマンド	目的
clear bgp all {*neighbor* | * | *as-number* | **peer-template** *name* | *prefix*} [**vrf** *vrf-name*]	すべてのアドレス ファミリから 1 つ以上のネイバーをクリアします。* を指定すると、すべてのアドレスファミリのすべてのネイバーが消去されます。引数は次のとおりです。 • *neighbor*：ネイバーの IPv4 または IPv6 アドレス。 • *as-number*：自律システム番号。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。 • *name*：ピア テンプレート名。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。 • *prefix*：IPv4 または IPv6 プレフィックス。そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。
clear bgp all dampening [**vrf** *vrf-name*]	すべてのアドレス ファミリのルート フラップ ダンプニング ネットワークをクリアします。*vrf-name* には最大 64 文字の英数字文字列を指定します。大文字と-小文字は区別されます。
clear bgp all flap-statistics [**vrf** *vrf-name*]	すべてのアドレス ファミリのルート フラップ統計情報をクリアします。*vrf-name* には最大 64 文字の英数字文字列を指定します。大文字と-小文字は区別されます。
clear bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} **dampening** [**vrf** *vrf-name*]	選択したアドレス ファミリのルート フラップ ダンプニング ネットワークをクリアします。*vrf-name* には最大 64 文字の英数字文字列を指定します。大文字と小文字は区別されます。
clear bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} **flap-statistics** [**vrf** *vrf-name*]	選択したアドレス ファミリのルート フラップ統計情報をクリアします。*vrf-name* には最大 64 文字の英数字文字列を指定します。大文字と-小文字は区別されます。

コマンド	目的
clear bgp {**ipv4** | **ipv6** } {*neighbor* |* | *as-number* | **peer-template** *name* | *prefix*} [**vrf** *vrf-name*]	選択したアドレス ファミリから 1 つ以上のネイバーをクリアします。* を指定すると、そのアドレス ファミリのすべてのネイバーが消去されます。引数は次のとおりです。 • *neighbor*：ネイバーの IPv4 または IPv6 アドレス。 • *as-number*：自律システム番号。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。 • *name*：ピア テンプレート名。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。 • *prefix*：IPv4 または IPv6 プレフィックス。そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

コマンド	目的
clear bgp {**ip** {**unicast** | **multicast**}} {*neighbor* |* |*as-number* | **peer-template** *name* | *prefix*} [**vrf** *vrf-name*]	1 つ以上のネイバーをクリアします。* を指定すると、そのアドレスファミリのすべてのネイバーが消去されます。引数は次のとおりです。 • *neighbor*：ネイバーの IPv4 または IPv6 アドレス。 • *as-number*：自律システム番号。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。 • *name*：ピア テンプレート名。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。 • *prefix*：IPv4 または IPv6 プレフィックス。そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。
clear bgp dampening [*ip-neighbor* | *ip-prefix*] [**vrf** *vrf-name*]	1 つ以上のネットワークのルート フラップ ダンプニングをクリアします。引数は次のとおりです。 • *ip-neighbor*：ネイバーの IPv4 アドレス。 • *ip-prefix*： IPv4 そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

コマンド	目的
clear bgp flap-statistics [*ip-neighbor* | *ip-prefix*] [**vrf** *vrf-name*]	1 つ以上のネットワークのルート フラップ統計情報をクリアします。引数は次のとおりです。 • *ip-neighbor*：ネイバーの IPv4 アドレス。 • *ip-prefix*： IPv4 そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。
clear ip mbgp {**ip** {**unicast** | **multicast**}} {*neighbor* | * | *as-number* | **peer-template** *name* | *prefix*} [**vrf** *vrf-name*]	1 つ以上のネイバーをクリアします。* を指定すると、そのアドレス ファミリのすべてのネイバーが消去されます。引数は次のとおりです。 • *neighbor*：ネイバーの IPv4 または IPv6 アドレス。 • *as-number*：自律システム番号。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。 • *name*：ピア テンプレート名。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。 • *prefix*：IPv4 または IPv6 プレフィックス。そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

コマンド	目的
clear ip mbgp dampening [*ip-neighbor* | *ip-prefix*] [**vrf** *vrf-name*]	1 つ以上のネットワークのルート フラップ ダンプニングをクリアします。引数は次のとおりです。 • *ip-neighbor*：ネイバーの IPv4 アドレス。 • *ip-prefix*： IPv4 そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。
clear ip mbgp flap-statistics [*ip-neighbor* | *ip-prefix*] [**vrf** *vrf-name*]	1 つ以上のネットワークのルート フラップ統計情報をクリアします。引数は次のとおりです。 • *ip-neighbor*：ネイバーの IPv4 アドレス。 • *ip-prefix*： IPv4 そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

ベーシック BGP の設定の確認

BGP の設定を表示するには、次のいずれかの作業を行います。

コマンド	目的
show bgp all [summary] [**vrf** *vrf-name*]	すべてのアドレス ファミリについて、BGP 情報を表示します。
show bgp convergence [**vrf** *vrf-name*]	すべてのアドレス ファミリについて、BGP 情報を表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix* **community** [**regexp** *expression* | [**community**] [**no-advertise**] [**no-export**] [**no-export-subconfed**]} [**vrf** *vrf-name*]	BGP コミュニティと一致する BGP ルートを表示します。
show bgp [**vrf** *vrf-name*] {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **community-list** *list-name* [**vrf** *vrf-name*]	BGP コミュニティ リストと一致する BGP ルートを表示します。

コマンド	目的
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix* **extcommunity** [**regexp** *expression* | [**generic** [**non-transitive** | **transitive**] *aa4:nn* [**exact-match**]} [**vrf** *vrf-name*]	BGP 拡張コミュニティと一致する BGP ルートを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix* **extcommunity-list** *list-name* [**exact-match**]} [**vrf** *vrf-name*]	BGP 拡張コミュニティ リストと一致する BGP ルートを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix* {**dampening dampened-paths** [**regexp** *expression*]} [**vrf** *vrf-name*]	BGP ルート ダンプニングの情報を表示します。ルート フラップ ダンプニング情報を消去するには、**clear bgp dampening** コマンドを使用します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix* **history-paths** [**regexp** *expression*] [**vrf** *vrf-name*]	BGP ルート ヒストリ パスを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix* **filter-list** *list-name* [**vrf** *vrf-name*]	BGP フィルタ リストの情報を表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **neighbors** [*ip-address* | *ipv6-prefix*] [**vrf** *vrf-name*]	BGP ピアの情報を表示します。これらのネイバーを消去するには、**clear bgp neighbors** コマンドを使用します。
show bgp {**ipv4** | **ipv6**} **unicast neighbors** [*ip-address* | *ipv6-prefix*] { [**advertised-routes** |**received-routes**] } [**detail**] [**vrf** *vrf-name*] **show bgp** {**ipv4** | **ipv6**} **unicast neighbors** [*ip-address* | *ipv6-prefix*] [**routes**] { [**advertised** |**received**] } [**detail**] [**vrf** *vrf-name*]	すべてのルートの詳細情報を表示します。 • インバウンドルートマップを評価する前にピアから受信しました 。 • アウトバウンドルートマップによって属性を更新する前に、ピアにアドバタイズされます。
show bgp {**ipv4** | **ipv6**} **unicast neighbors** [*ip-address* | *ipv6-prefix*] [**routes**] [**detail**] [**vrf** *vrf-name*]	インバウンド ルート マップの評価後に、このピアから受信したすべてのルートの詳細情報を表示します。
show bgp {**ipv4** | **ipv6**} **unicast neighbors** [*ip-address* | *ipv6-prefix*] [**advertised-routes processed**] [**vrf** *vrf-name*]	**処理された**オプションを使用してアウトバウンド ルート マップによってパス属性を更新した後、ピアにアドバタイズされたすべてのルートの要約情報を表示します。
show bgp {**ipv4** | **ipv6**} **unicastneighbors** [*ip-address* | *ipv6-prefix*] [**advertised-routes processed**] [**detail**] [**vrf** *vrf-name*]	**処理された**オプションを使用してアウトバウンド ルート マップによってパス属性を更新した後、ピアにアドバタイズされたすべてのルートの詳細情報を表示します。

コマンド	目的
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **neighbors** [*ip-address* | *ipv6-prefix*] {**nexthop** | **nexthop-database**} [**vrf** *vrf-name*]	BGP ルート ネクスト ホップの情報を表示します。
show bgp paths	BGP パス情報を表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **policy** *name* [**vrf** *vrf-name*]	BGP ポリシー情報を表示します。ポリシー情報を消去するには、**clear bgp polic** コマンドを使用します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **prefix-list** *list-name* [**vrf** *vrf-name*]	プレフィックス リストと一致する BGP ルートを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **received-paths** [**vrf** *vrf-name*]	ソフト再構成用に保管されている BGP パスを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **regexp** *expression* [**vrf** *vrf-name*]	AS_path 正規表現と一致する BGP ルートを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **route-map** *map-name* [**vrf** *vrf-name*]	ルート マップと一致する BGP ルートを表示します。
show bgp peer-policy *name* [**vrf** *vrf-name*]	BGP ピア ポリシー情報を表示します。
show bgp peer-session *name* [**vrf** *vrf-name*] show bgp peer-session	BGP ピア セッション情報を表示します。
show bgp peer-template *name* [**vrf** *vrf-name*]	BGP ピアテンプレート情報を表示します。ピアテンプレートのすべてのネイバーを消去するには、**clear bgp peer-template** コマンドを使用します。
show bgp process	BGP プロセス情報を表示します。
show {**ipv** | **ipv6**} **bgp** [*options*]	BGP のステータスと構成情報を表示します。
show {**ipv** | **ipv6**} **mbgp** [*options*]	BGP のステータスと構成情報を表示します。
show running-configuration bgp	現在実行中の BGP コンフィギュレーションを表示します。

BGP 統計情報のモニタリング

BGP の統計情報を表示するには、次のコマンドを使用します。

コマンド	目的
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **flap-statistics** [**vrf** *vrf-name*]	BGP ルート フラップの統計情報を表示します。これらの統計情報をクリアするには、**clear bgp flap-statistics command** を使用します。
show bgp sessions [**vrf** *vrf-name*]	すべてのピアの BGP セッションを表示します。これらの統計情報をクリアするには、**clear bgp sessions** コマンドを使用します。
show bgp statistics	BGP 統計情報を表示します。

ベーシック BGP の設定例

次に、ベーシック BGP 設定の例を示します。

```
switch(config)# feature bgp
switch(config)# router bgp 64496
switch(config-router)# neighbor 2001:ODB8:0:1::55 remote-as 64496
switch(config-router)# address-family ipv6 unicast
switch(config-router-af)# next-hop-self
```

関連項目

BGP の関連項目は、次のとおりです。

- 高度な BGP の設定 （357 ページ）
- Route Policy Manager の設定 （587 ページ）

次の作業

次の機能の詳細については、高度な BGP の設定 （357 ページ）を参照してください。

- ピア テンプレート
- ルートの再配布
- ルート マップ

その他の参考資料

BGP の実装に関連する詳細情報については、次の項を参照してください。

ベーシック BGP の MIB

MIB	MIB のリンク
BGP に関連する MIB	MIB を検索およびダウンロードするには、次の URL にアクセスしてください。 ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

第 11 章

高度な BGP の設定

この章は、次の項で構成されています。

- 拡張 BGP について （358 ページ）
- 拡張 BGP の前提条件 （372 ページ）
- 拡張 BGP に関する注意事項と制限事項 （372 ページ）
- デフォルト設定 （378 ページ）
- 高度な BGP の設定 （378 ページ）
- BGP 追加パスの設定 （400 ページ）
- eBGP の設定 （405 ページ）
- AS 連合の設定 （410 ページ）
- ルート リフレクタの設定 （411 ページ）
- アウトバウンド ルート マップを使用した、反映されたルートのネクスト ホップの設定 （414 ページ）
- ルート ダンプニングの設定 （417 ページ）
- ロード シェアリングおよび ECMP の設定 （417 ページ）
- BGP 経由不等コストマルチパス（UCMP） （418 ページ）
- UCMP over BGP の有効化 （418 ページ）
- BGP 経由 UCMP の注意事項と制限事項 （419 ページ）
- 最大プレフィックス数の設定 （419 ページ）
- DSCP の設定 （420 ページ）
- ダイナミック機能の設定 （420 ページ）
- 集約アドレスの設定 （421 ページ）
- BGP ルートの抑制 （422 ページ）
- BGP 条件付きアドバタイズメントの設定 （423 ページ）
- ルートの再配布の設定 （426 ページ）
- DMZ リンク帯域幅 （427 ページ）
- デフォルト ルートのアドバタイズ （439 ページ）
- BGP 属性フィルタリングの設定とエラー処理 （440 ページ）
- BGP の調整 （443 ページ）
- ポリシーベースのアドミニストレーティブ ディスタンスの設定 （449 ページ）

• マルチプロトコル BGP の設定 （451 ページ）
• BMP の設定 （452 ページ）
• BGP ローカル ルート リーク （455 ページ）
• BGP グレースフル シャットダウン （464 ページ）
• グレースフル リスタートの設定 （477 ページ）
• 仮想化の設定 （480 ページ）
• 拡張 BGP の設定の確認 （482 ページ）
• BGP 統計情報のモニタリング （484 ページ）
• 設定例 （485 ページ）
• 関連項目 （486 ページ）
• その他の参考資料 （486 ページ）

拡張 BGP について

BGP は、組織または自律システム間のループフリールーティングを実現する、インタードメインルーティング プロトコルです。Cisco NX-OS は BGP バージョン 4 をサポートしています。BGP v4 に組み込まれているマルチプロトコル拡張機能を使用すると、IP マルチキャストルートおよび複数のレイヤ 3 プロトコル アドレス ファミリに関するルーティング情報を BGP に伝送させることができます。BGP では、他の BGP 対応デバイス（BGP ピア）との間で TCP セッションを確立するために、信頼できるトランスポート プロトコルとして TCP を使用します。外部組織に接続するときには、ルータが外部 BGP（eBGP）ピアリング セッションを作成します。同じ組織内の BGP ピアは、内部 BGP（iBGP）ピアリング セッションを通じて、ルーティング情報を交換します。

ピア テンプレート

BGP ピア テンプレートを使用すると、類似した BGP ピア間で再利用できる共通のコンフィギュレーション ブロックを作成できます。各ブロックでは、ピアに継承させる一連の属性を定義できます。継承した属性の一部を上書きすることもできるので、非常に柔軟性のある方法で、繰り返しの多い BGP の設定を簡素化できます。

Cisco NX-OS は、3 種類のピア テンプレートを実装します。

- peer-session テンプレートでは、トランスポートの詳細、ピアのリモート自律システム番号、セッション タイマーなど、BGP セッション属性を定義します。peer-session テンプレートは、別の peer-session テンプレートから属性を継承することもできます（ローカル定義の属性によって、継承した peer-session 属性は上書きされます）。

- peer-policy テンプレートでは、着信ポリシー、発信ポリシー、フィルタ リスト、プレフィックス リストを含め、アドレス ファミリに依存する、ピアのポリシー要素を定義します。peer-policy テンプレートは、一連の peer-policy テンプレートからの継承が可能です。Cisco NX-OS は、継承設定のプリファレンス値で指定された順序で、これらの peer-policy テンプレートを評価します。最小値が大きい値よりも優先されます。

- peer テンプレートは、peer-session および peer-policy テンプレートからの継承が可能であり、ピアの定義を簡素化できます。peer テンプレートの使用は必須ではありませんが、peer テン

プレートによって再利用可能なコンフィギュレーションブロックが得られるので、BGP の設定を簡素化できます。

認証

BGP ネイバーセッションに認証を設定できます。この認証方式によって、ネイバーに送られる各 TCP セグメントに MD5 認証ダイジェストが追加され、不正なメッセージや TCP セキュリティ アタックから BGP が保護されます。

（注） MD5 パスワードは、BGP ピア間で一致させる必要があります。

ルート ポリシーおよび BGP セッションのリセット

BGP ピアにルート ポリシーを関連付けることができます。ルート ポリシーではルート マップを使用して、BGP が認識するルートを制御または変更します。着信または発信ルート アップデートに関するルート ポリシーを設定できます。ルート ポリシーはプレフィックス、AS_path 属性など、さまざまな条件で一致が必要であり、ルートを選択して受け付けるかまたは拒否します。ルート ポリシーでパス属性を変更することもできます。

BGP ピアに適用するルート ポリシーを変更する場合は、そのピアの BGP セッションをリセットする必要があります。Cisco NX-OS は、BGP セッションをリセットするため、次の 3 つのメカニズムをサポートしています。

- ハード リセット：ハード リセットでは、指定されたピアリング セッションが TCP 接続を含めて切断され、指定のピアからのルートが削除されます。このオプションを使用すると、BGP ネットワーク上のパケット フローが中断します。ハードリセットは、デフォルトでディセーブルです。
- ソフト再構成着信：ソフト再構成着信によって、セッションをリセットすることなく、指定されたピアのルーティング アップデートが開始されます。このオプションを使用できるのは、着信ルート ポリシーを変更する場合です。ソフト再構成着信の場合、ピアから受け取ったすべてのルートのコピーを保存したあとで、着信ルート ポリシーを介してルートが処理されます。着信ルート ポリシーを変更する場合、Cisco NX-OS は変更された着信ルート ポリシーを介して保存ルートを渡し、既存のピアリング セッションを切断することなく、ルート テーブルをアップデートします。ソフト再構成着信の場合、まだフィルタリングされていない BGP ルートの保存に、大量のメモリ リソースを使用する可能性があります。ソフト再構成着信は、デフォルトでディセーブルです。
- ルート リフレッシュ：ルート リフレッシュでは、着信ルート ポリシーの変更時に、サポートするピアにルート リフレッシュ要求を送信することによって、着信ルーティング テーブルがダイナミックにアップデートされます。リモート BGP ピアは新しいルート コピーで応答し、ローカル BGP スピーカが変更されたルート ポリシーでそれを処理します。Cisco NX-OS は自動的に、プレフィックスのアウトバウンド ルートの更新をピアに送信します。

- BGP ピアは、BGP ピア セッションの確立時に、BGP 機能ネゴシエーションの一部として、ルート リフレッシュ機能をアドバタイズします。ルート リフレッシュは優先オプションであり、デフォルトでイネーブルです。

（注） BGP はさらに、ルート再配布、ルート集約、ルート ダンプニングなどの機能にルート マップを使用します。ルート マップの詳細については、Route Policy Manager の設定（587 ページ）を参照してください。

eBGP

eBGP を使用すると、異なる AS からの BGP ピアを接続し、ルーティング アップデートを交換できます。外部ネットワークへの接続によって、自分のネットワークから他のネットワークへ、またインターネットを介して、トラフィックを転送できます。

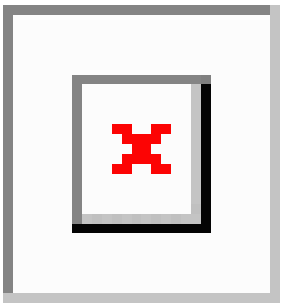
通常、eBGP ピアリングは、インターフェイスがダウンしたときにコンバージェンスが高速になるように、直接接続されたインターフェイス上で行う必要があります。

iBGP

iBGP を使用すると、同じ自律システム内の BGP ピアを接続できます。iBGP はマルチホーム BGP ネットワーク（同じ外部自律システムに対して複数の接続があるネットワーク）に使用できます。

図に、大きい BGP ネットワークの中の iBGP ネットワークを示します。

図 *29 : iBGP* ネットワーク



iBGP ネットワークはフルメッシュです。各 iBGP ピアは、ネットワーク ループを防止するために、他のすべての iBGP ピアに対して直接接続されています。

ネイバー コンフィギュレーション モードで update-source が設定された単一ホップ iBGP ピアでは、ピアは高速外部フェールオーバーをサポートします。

iBGP ピアリング セッションの確立には、ループバック インターフェイスを使用します。ループバック インターフェイスは、インターフェイス フラップが発生する可能性が小さいからです。インターフェイス フラップが発生するのは、障害またはメンテナンスが原因で、インターフェイスが管理上アップまたはダウンになったときです。マルチホップ、高速外部フォールオーバー、AS パス属性のサイズ制限については、eBGP の設定（405 ページ）セクションを参照してください。

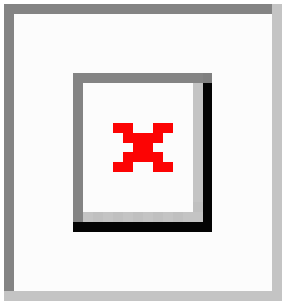
（注） iBGP ネットワークでは別個のインテリア ゲートウェイ プロトコルを設定する必要があります。

AS 連合

フルメッシュの iBGP ネットワークは、iBGP ピア数が増えるにしたがって複雑になります。自律システムを複数のサブ自律システムに分割し、それを 1 つの連合としてまとめることによって、iBGP メッシュを緩和できます。連合は、同じ自律システム番号を使用して外部ネットワークと通信する、iBGP ピアからなるグループです。各サブ AS はその中ではフルメッシュであり、同じ連合内の他のサブ AS に対する少数の接続があります。

図に BGP ネットワークが 2 つのサブ AS と 1 つの連合に分けられて表示されます。

図 *30 : AS* 連合



この例では、AS10 が 2 つの AS（AS1 および AS2）に分割されています。各サブ AS はフルメッシュですが、サブ AS 間のリンクは 1 つだけです。AS 連合を使用することによって、 のフルメッシュ AS に比べて、リンク数を少なくできます。

ルート リフレクタ

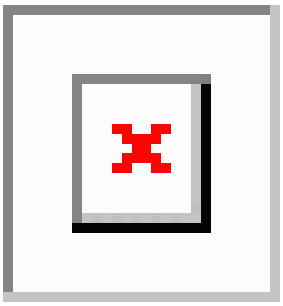
すべての iBGP ピアが完全に一致する必要がないように、ルート リフレクタが学習したルートをネイバーに渡すルート リフレクタ構成を使用することによって、iBGP メッシュを削減できます。

ある iBGP ピアをルート リフレクタとして設定すると、そのピアが iBGP で学習したルートを一連の iBGP ネイバーに渡す役割を担います。

図に、メッシュの iBGP スピーカを 4 つ（ルータ A、B、C、D）使用する、単純な iBGP 構成を示します。ルート リフレクタを使用しなかった場合、外部ネイバーからルートを受け取ったルータ A は、3 つの iBGP ネイバーのすべてにルートをアドバタイズします。

図では、ルータ B がルート リフレクタです。ルート リフレクタは、ルータ A からアドバタイズされたルートを受信すると、ルータ C と D へのルートをアドバタイズ（リフレクト）します。ルータ A は、ルータ C と D の両方にアドバタイズする必要がなくなります。

図 *31* : ルート リフレクタ



ルート リフレクタおよびそのクライアント ピアは、クラスタを形成します。ルート リフレクタのクライアント ピアとして動作するように、すべての iBGP ピアを設定する必要はありません。ただし、完全な BGP アップデートがすべてのピアに届くように、非クライアント ピアはフルメッシュとして設定する必要があります。

機能ネゴシエーション

BGP スピーカは機能ネゴシエーション機能を使用することによって、ピアでサポートされている BGP 拡張機能を学習できます。機能ネゴシエーションによって、リンクの両側の BGP ピアがサポートする機能セットだけを BGP に使用させることができます。

BGP ピアが機能ネゴシエーションをサポートしない場合で、なおかつアドレス ファミリが IPv4 として設定されている場合、Cisco NX-OS は機能ネゴシエーションを行わずに、ピアとの新規セッションを試みます。他のマルチプロトコル設定（IPv6 など）の場合は、機能ネゴシエーションが不可欠です。

ルート ダンプニング

ルートダンプニングは、インターネットワーク上でのフラッピングルートの伝搬を最小限に抑える BGP 機能です。ルート フラップが発生するのは、使用可能ステートと使用不能ステートが短時間で次々切り替わる場合です。

AS1、AS2、および AS3 という 3 つの BGP 自律システムからなるネットワークの場合について考えてみます。AS1 のルートがフラップした（使用不能になった）とします。ルート ダンプニングを使用しない場合、AS1 は AS2 に回収メッセージを送信します。AS2 は AS3 にその回収メッセージを伝達します。フラッピング ルートが再び発生すると、AS1 から AS2 にアドバタイズメント メッセージを送信し、AS2 は AS3 にそのアドバタイズメントを送信します。ルートの使用不能と使用可能が繰り返されると、AS1 は多数の回収メッセージおよびアドバタイズメント メッセージを送信することになり、それが他の自律システムに伝播します。

ルートダンプニングによって、フラッピングを最小限に抑えることができます。ルートフラップが発生したとします。（ルート ダンプニングがイネーブルの）AS2 がルートにペナルティとして 1000 を割り当てます。AS2 は引き続き、ネイバーにルートの状態をアドバタイズします。ルート フラップが発生するたびに、AS2 がペナルティ値を追加します。ルート フラップが頻繁に発生して、ペナルティが設定可能な抑制限度を超えると、AS2 はフラップ回数に関係なく、ルートのアドバタイズを中止します。その結果、ルートが減衰（ダンプニング）します。

ルートに与えられたペナルティは、再使用限度に達するまで減衰します。その時点で、AS2 は再びルートをアドバタイズします。再使用限度が 50% になると、AS2 はそのルートのダンプニング情報を削除します。

（注） ルート ダンプニングがイネーブルの場合は、ピアのリセットによってルートが回収されても、リセット中の BGP にはペナルティは適用されません。

ロード シェアリングおよびマルチパス

BGP はルーティング テーブルに、同じ宛先プレフィックスに到達する複数の等コスト eBGP または iBGP パスを組み込むことができます。その場合、宛先プレフィックスへのトラフィックは、組み込まれたすべてのパス間で共有されます。

as-path multipath-relax コマンドを効果的に構成するには、BGP で VRF ごとに コマンドを構成します。また、複数のルータがカスタム VRF ルートターゲット（RT）にインストールされるように、カスタム VRF で as-path multipath-relax コマンドを構成します。

BGP ベストパス アルゴリズムでは、次の属性が同じ場合に、等コスト パスと見なされます。

- 重量
- ローカル プリファレンス
- AS_path
- オリジン コード
- Multi-Exit Discriminator（MED）
- BGP ネクスト ホップまでの IGP コスト

BGP はこれら複数のパスの中から、ベスト パスとして 1 つだけ選択し、そのパスを BGP ピアにアドバタイズします。詳細については、「BGP追加パス」の項を参照してください。

（注） 異なる AS 連合から受け取ったパスは、外部 AS_path 値およびその他の属性が同じ場合に、等コスト パスと見なされます。

（注） iBGP マルチパスに関してルート リフレクタを設定すると、ルート リフレクタが、選択されたベスト パスをピアにアドバタイズします。そのパスのネクスト ホップは変更されません。

BGP の追加パス

1 つの BGP 最良パスだけがアドバタイズされ、BGP スピーカは特定ピアからの特定プレフィックスの 1 パスだけを受け入れます。BGP スピーカが同じセッション内で同じプレフィックスの複数のパスを受信した場合、最新のアドバタイズメントを使用します。

BGP は、以前のパスに代わる新しいパスなしで、BGP スピーカが同じプレフィックスに対して複数のパスを伝播し、受け入れることを可能にする追加のパス機能をサポートします。この機能は、BGP スピーカのピアが、プレフィックスごとの複数パスのアドバタイズおよび受信をサポートし、また、そのパスのアドバタイズをサポートするかどうかネゴシエートすることを可能にします。特別な 4 バイトのパス ID は、ピア セッションを介して送信される同じプレフィックスに対して複数のパスを区別するため、ネットワーク層到達可能性情報（NLRI）に追加されます。次の図に、追加の BGP パス機能を示します。

図 *32*：追加パスの機能を持つ *BGP* ルート アドバタイズメント

BGP 追加パス設定の詳細については、BGP 追加パスの設定（400 ページ）の項を参照してください。

ルート集約

集約アドレスを設定できます。ルート集約を使用すると、固有性の強い一連のアドレスをすべての固有アドレスを代表する 1 つのアドレスに置き換えることによって、ルート テーブルを簡素化できます。たとえば、10.1.1.0/24、10.1.2.0/24、および 10.1.3.0/24 という固有性の強い 3 つのアドレスを 1 つの集約アドレス 10.1.0.0/16 に置き換えることができます。

アドバタイズされるルートが少なくなるように、BGP ルートテーブル内には集約プレフィックスが存在します。

（注） Cisco NX-OS は、自動ルート集約をサポートしません。

ルート集約はフォワーディング ループにつながる可能性があります。この問題を回避するために、集約アドレスのアドバタイズメントを生成するときに、BGP はローカル ルーティング テーブルに、その集約アドレスに対応するサマリー廃棄ルートを自動的に組み込みます。BGP はサマリー廃棄のアドミニストレーティブ ディスタンスを 220 に設定し、ルート タイプを廃棄に設定します。BGP はネクストホップ解決に廃棄ルートを使用しません。

ユーザが **aggregate-address** コマンドを発行すると、BGP テーブルにサマリー エントリが作成されますが、サマリー エントリは、集約のサブセットがテーブルで見つかるまでアドバタイズできません。

BGP 条件付きアドバタイズメント

BGP条件付きアドバタイズメントを使用すると、プレフィックスがBGPテーブルに存在するかどうかに基づいてルートをアドバタイズまたは撤回するようにBGPを設定できます。この機能は、たとえば、BGPでいずれかのプロバイダーにプレフィックスをアドバタイズするようなマルチホーム ネットワーク（他のプロバイダーからの情報が存在しない場合のみ）で便利です。

AS1、AS2、および AS3 という 3 つの BGP 自律システムからなるネットワークの例について考えてみます。この例で、AS1 と AS3 はインターネットと AS2 に接続しています。条件付きアドバタイズメントを使用しない場合、AS2 はすべてのルートを AS1 と AS3 の両方にプロパゲートします。条件付きアドバタイズメントを使用すれば、AS1 からのルートが存在しない場合のみ（たとえば AS1 へのリンクがダウンした場合）、特定のルートを AS3 にアドバタイズするように AS2 を設定できます。

BGP条件付きアドバタイズメントでは、設定されたルートマップに一致する各ルートに、存在テストまたは非存在テストが追加されます。「BGP 条件付きアドバタイズメントの設定」を参照してください。

BGP ネクスト ホップ アドレス トラッキング

BGPは、インストールされているルートのネクストホップアドレスをモニタして、ネクストホップの到達可能性の確認、および BGP ベスト パスの選択、インストール、検証を行います。BGP ネクストホップ アドレスのトラッキングを行うと、ネクストホップの到達可能性に影響を及ぼす可能性のあるルート変更がルーティング情報ベース（RIB）で行われたときに確認プロセスをトリガーすることで、このようなネクストホップ到達可能性テストの速度が向上します。

ネクスト ホップ情報が変更されると、BGP は RIB から通知を受信します（イベント駆動型の通知）。BGP は、次のいずれかのイベントが発生したときに通知を受けます。

- ネクスト ホップが到達不能になった。
- ネクスト ホップが到達可能になった。
- ネクスト ホップへの完全再帰のインテリア ゲートウェイ プロトコル（IGP）メトリックが変更された。
- ファーストホップの IP アドレスまたはファーストホップのインターフェイスが変更された。
- ネクスト ホップが接続された。
- ネクスト ホップが接続解除された。
- ネクスト ホップがローカル アドレスになった。
- ネクスト ホップが非ローカル アドレスになった。

（注） 到達可能性および再帰メトリック イベントは、最適パスの再計算をトリガーします。

RIB からのイベント通知は、クリティカルおよび非クリティカルとして分類されます。クリティカルおよび非クリティカル イベントの通知は、別々のバッチで送信されます。ただし、非クリティカルイベントが保留中であり、クリティカルイベントを読み込む要求がある場合は、非クリティカル イベントがクリティカル イベントとともに送信されます。

- クリティカルなイベントとは、異なるパスに対してスイッチオーバーの原因となるネクスト ホップの消失など、ネクスト ホップの到達可能性に関連しています。異なるパスに対してスイッチオーバーの原因となるネクスト ホップの IGP メトリックの変更は、クリティカルなイベントと見なすことができます。
- 非クリティカルなイベントとは、最適パスに影響を与えたり、単一のネクスト ホップに IGP メトリックを変更したりせずに追加されるネクスト ホップに関連しています。

詳細については、「BGP ネクスト ホップ アドレス トラッキングの設定」を参照してください。

ルートの再配布

スタティック ルートまたは他のプロトコルからのルートを再配布するように、BGP を設定できます。再配布を指定したルート マップを設定して、どのルートが BGP に渡されるかを制御する必要があります。ルート マップを使用すると、宛先、送信元プロトコル、ルート タイプ、ルート タグなどの属性に基づいて、ルートをフィルタリングできます。詳細については、Route Policy Manager の設定 （587 ページ）を参照してください。

ルートマップを使用して両シナリオのデフォルト動作を無効にできますが、ルートマップの正しくない使用によってネットワーク ループが発生することがあるため、そうする場合は注意が必要です。次に、デフォルトの動作の変更にルート マップを使用する例を示します。

ルート マップの変更によって、シナリオ 1 のデフォルトの動作を次のように変更できます。

```
route-map foo permit 10
   match route-type internal
router ospf 1
   redistribute bgp 100 route-map foo
```

同様に、ルート マップの変更によって、シナリオ 2 のデフォルトの動作を次のように変更できます。

```
route-map foo deny 10
  match route-type internal
router ospf 1
   vrf bar
     redistribute bgp 100 route-map foo
```

ラベル付きユニキャスト ルートとラベルなしユニキャスト ルート

リリース7.0(3)I7(6) では、SAFI-1（ラベルなしユニキャスト）およびSAFI-4（ラベル付きユニキャスト ルーティング）が単一セッションの IPv4 BGP でサポートされるようになりました。詳細については、『*Cisco Nexus 9000 Series NX-OS Label Switching Configuration Guide*、*Release 7.x*』を参照してください。

BFD

この機能では、IPv4 および IPv6 用の双方向フォワーディング検出（BFD）をサポートします。BFD は、転送パスの障害を高速で検出することを目的にした検出プロトコルです。BFD は 2 台の隣接デバイス間のサブセカンド障害を検出し、BFD の負荷の一部を、サポートされるモジュール上のデータ プレーンに分散できるため、プロトコル hello メッセージよりも CPU を使いません。

BGP の BFD は eBGP ピアおよび iBGP シングルホップ ピアでサポートされます。BFD を使用している iBGP シングル ホップ ピアのネイバー コンフィギュレーション モードで **update-source** オプションを設定します。

Cisco NX-OS リリース 9.3(3) 以降では、BGP の BFD は BGP IPv4 と IPv6 のプレフィックス ピアでもサポートされます。このサポートにより、BGP はマルチホップ BFD を使用できるようになり、BGP コンバージェンス時間が改善されます。プレフィックス ピアでは、シングルホップ BGP とマルチホップ BGP の両方がサポートされます。

Cisco NX-OS リリース 9.3(3) 以降、BFD は IPv4 および IPv6 アドレス ファミリの IPv6 リンクローカルを介した BGP インターフェイスピアリングをサポートします。ただし、BFD マルチホップはアンナンバード BGP ではサポートされません。

詳細については、『Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide』を参照してください。

BGP の調整

BGP タイマーによって、さらにベストパス アルゴリズムの調整によって、BGP のデフォルト動作を変更できます。

BGP タイマー

BGP では、ネイバー セッションおよびグローバル プロトコル イベントにさまざまなタイプのタイマーを使用します。確立されたセッションごとに、最低限 2 つのタイマーがあります。定期的にキープアライブ メッセージを送信するためのタイマー、さらに想定時間内にピアのキープアライブが届かなかった場合に、セッションをタイムアウトさせるためのタイマーです。また、個々の機能を処理するための、その他のタイマーがあります。これらのタイマーは通常、秒単位で設定します。タイマーには、異なる BGP ピアで同じタイマーが異なるタイミングでスタートするように、ランダム アジャストメントが組み込まれています。

ベストパス アルゴリズムの調整

オプションの設定パラメータによって、ベストパス アルゴリズムのデフォルト動作を変更できます。たとえば、アルゴリズムでの Multi-Exit Discriminator（MED）属性およびルータ ID の扱い方を変更できます。

マルチプロトコル BGP

Cisco NX-OS の BGP は、複数のアドレス ファミリをサポートします。マルチプロトコル BGP（MP-BGP）は、アドレス ファミリに応じて異なるルート セットを伝送します。BGP ではたとえば、IPv4 ユニキャスト ルーティング用のルート セットを 1 つ、IPv4 マルチキャスト ルーティング用のルート セットを 1 つ、さらに IPv6 マルチキャスト ルーティング用のルート セットを 1 つ伝送できます。IP マルチキャスト ネットワークではリバース パス フォワーディング（RPF）のチェックに MP-BGP を使用できます。

（注） マルチキャスト BGP ではマルチキャスト状態情報をプロパゲートしないため、プロトコル独立マルチキャスト（PIM）などのマルチキャスト プロトコルが必要です。

マルチプロトコル BGP 設定をサポートするには、ルータ アドレスファミリおよびネイバー アドレス ファミリの各コンフィギュレーション モードを使用します。MP-BGP では、設定されたアドレス ファミリごとに別々の RIB が維持されます（ユニキャスト RIB と、BGP のマルチキャスト RIB など）。

マルチプロトコル BGP ネットワークは下位互換性がありますが、マルチプロトコル拡張機能をサポートしない BGP ピアは、アドレス ファミリ ID 情報など、マルチプロトコル拡張機能が伝送するルーティング情報を転送できません。

RFC 5549

BGP は RFC 5549 をサポートしており、IPv4 プレフィックスを IPv6 ネクスト ホップで伝送できます。BGP はすべてのホップで実行されるため、すべてのルータが IPv4 および IPv6 トラフィックを転送できます。したがって、ルータ間で IPv6 トンネルをサポートする必要はありません。BGP は、IPv6 ルートを介した IPv4 を Unicast Route Information Base（URIB）にインストールします。

Cisco NX-OS リリース9.2(2) 以降では、-R タイプのライン カードを搭載した Cisco Nexus 9500 プラットフォーム スイッチは、RFC 5549 をサポートします。

現在、NX-OS は IPv4 ルートの IPv6 再帰ネクストホップ（RNH）をサポートしていません。

RFC 6368

はじめに

このセクションでは、Cisco NX-OS のプロバイダーエッジ（PE）機能とカスタマーエッジ（CE）機能間で内部ボーダー ゲートウェイ プロトコル（iBGP）がどのように実装されているかについて説明します。

現在の展開で、プロバイダー/カスタマー エッジのルーティング プロトコルとして BGP を使用すると、VPN プロバイダー自律システム（AS）とカスタマー ネットワーク自律システム間の外部ピアリングとしてピアリング セッションが設定されます。

RFC 6368では、これらのピアがiBGPピアとして設定されるようになりました。

Cisco NX-OSリリース10.1（2）以降では、EVPN-VxLANv4およびEVPN-VxLANv6のRFC 6368サポートが有効になっています。

フレームワーク

Cisco NX-OSリリース10.1（2）以降では、iBGP PE-CE機能を導入しています。

- as-override を使用した外部 Border Gateway Protocol（eBGP）を展開せずに、VRF の複数のサイトで単一の自律システム番号（ASN）を持つことができます。
- プロバイダー コアがまるで 1 つの透過ルート リフレクタ（RR）のように機能する、CE ルータへの内部ルート リフレクションを提供したいと考えます。

この機能を使用VRF サイトは、プロバイダー コアと同じ ASN を持つことができます。ただし、VRF サイトの ASN が プロバイダー コアの ASN と異なっている場合は、この機能のローカル自律システム（AS）を使用して、同じであるように表示できます。

iBGP PE-CE の実装

この機能を動作させるのは、次の 2 つの主要部分です。

- プロバイダー コアで VPN BGP 属性を透過的に伝送するために、新しい属性である `ATTR_SET` が BGP プロトコルに追加されました。
- PE ルータを、VRF 内の CE ルータへの iBGP セッションの RR にします。

新しい `ATTR_SET` 属性ではプロバイダーがカスタマーの BGP 属性すべてを透過的に伝送でき、プロバイダー属性や BGP ポリシーに干渉することがありません。こうした属性にはクラスタ リスト、ローカル設定などがあります。

BGP カスタマー ルート属性

`ATTR_SET` は、プロバイダー カスタマーの VPN BGP 属性を伝送するために使用される、新しい BGP 属性です。これは過渡的なオプション属性です。この属性では、Local Preference、Med、Origin、AS Path、Originator ID、Cluster list属性がプロバイダーネットワーク全体で伝送されます。`ATTR_SET` 属性の形式は次のとおりです。

```
+-----------------------------+
| Attr Flags□O | T□Code = 128  |
+-----------------------------+
| Attr. Length (1 or 2 octets) |
+-----------------------------+
| Origin AS (4 octets)        |
+-----------------------------+
|Path Attributes (variable)    |
+-----------------------------+
```

- 属性フラグは、通常の BGP 属性フラグです。
- 属性の長さは、この属性の長さが 1 オクテットであるか 2 オクテットであるかを示します。

- Origin AS フィールドある AS で発生するルートが、適切な AS_PATH 操作を行われずに、別の AS にリークされないようにします。
- 可変長-のパス属性フィールドには、プロバイダーコアで伝送されなければならない VPN BGP 属性が含まれます。

iBGP PE-CE の実装の詳細については、「iBGP PE-CE 機能の IOS 実装」を参照してください。

次に、iBGPカスタマーエッジデバイスのPEデバイスでのBGPネイバー設定の例を示します。

```
router bgp 200
vrf nxbgp3-leaf2-2
address-family ipv4 unicast
redistribute static route-map ALLOW-ALL
address-family ipv6 unicast
redistribute static route-map ALLOW-ALL
neighbor 101.101.101.101 remote-as 200
description ibgp sample config
internal-vpn-client (1)
address-family ipv4 unicast
route-reflector-client (2)
next-hop-self (3)
```

BGP モニタリング プロトコル

BGP モニタリング プロトコル（BMP）は、BGP アップデートとピア統計情報をモニタし、すべての Cisco Nexus 9000 シリーズ スイッチでサポートされます。

このプロトコルを使用して、BGP スピーカーは外部 BMP サーバに接続し、BGP イベントに関する情報を送信します。1つの BGP スピーカーに最大 2 つの BMP サーバを設定でき、各 BGP ピアはBMP サーバのすべてまたはサブセットによるモニタリング用に設定できます。BGP スピーカーは、BMP サーバからの情報を受け入れません。

グレースフル リスタートおよびハイ アベイラビリティ

Cisco NX-OS は、BGP に対してノンストップ フォワーディングとグレースフル リスタートをサポートしています。

BGP ルーティング プロトコル情報がフェールオーバー後に復元されている間に、転送情報ベース（FIB）内の既知のルートでデータ パケットを転送するように、BGP の無停止フォワーディング（NSF）を使用できます。NSF では、BGP ピアはルーティング フラップと無縁です。フェールオーバー時に、データ トラフィックはインテリジェントモジュール経由で転送され、スタンバイ スーパーバイザがアクティブになります。

Cisco NX-OS ルータでコールド リブートが発生した場合、ネットワークはルータへのトラフィック転送を中止し、ネットワーク トポロジからルータを削除します。この状況では、BGP は非グレースフル リスタートになり、すべてのルートが削除されます。Cisco NX-OS がスタートアップ コンフィギュレーションを適用すると、BGP はピアリング セッションを再び確立して、ルートを再学習します。

Cisco NX-OS デュアル スーパーバイザ構成のルータでは、ステートフル スーパーバイザ スイッチオーバーが実行されます。スイッチオーバーの間、BGP は無停止フォワーディングを使用し、FIB

の情報に基づいてトラフィックを転送します。システムがネットワーク トポロジから取り除かれることはありません。ネイバーが再起動しているルータは、「ヘルパー」と呼ばれます。スイッチオーバー後、グレースフル リスタート動作が開始されます。この処理が進行中の際、2 つのルータはネイバー関係を再確立し、これらの BGP ルートを交換します。それらネイバー関係が再起動したとしても、ヘルパーは再起動中のピアを指すプレフィックスを転送し続け、再起動中のルータはピアへトラフィックを転送し続けます。再起動中のルータがグレースフル リスタート可能なすべての BGP ピアを持つ場合、グレースフル リスタートが完了し、BGP は再び動作可能なネイバーを通知します。

BGP は、グレースフル リスタート タイマーが期限切れになる前に収束する必要があります。一時的なトラフィック損失を回避するために、ルート スケールの大きいネットワークでは、それに応じて BGP グレースフル リスタート タイマーを増やす必要があります。BGP 自体が他の BGP セッションを開くための到達可能性を提供する場合は、最初のアンダーレイ セッションがすでに収束した後にオーバーレイセッションを収束させるために必要な追加の時間に対応するため、stalepath-time も増やす必要があります。

グレースフル リスタート動作中であることがルータで検出されると、両方のルータがそれぞれのトポロジ テーブルを交換します。すべての BGP ピアからルート アップデートを受信したルータは、古いルートをすべて削除し、アップデートされたルートでベストパス アルゴリズムを実行します。

スイッチオーバーが完了すると、Cisco NX-OS は実行コンフィギュレーションを適用し、BGP は自身が再度使用可能になったことをネイバーに通知します。

ネイバー コンフィギュレーション モードで update-source が設定された単一ホップ iBGP ピアでは、ピアは高速外部フェールオーバーをサポートします。

Cisco NX-OS リリース 9.3(3) 以降、BGP プレフィックス ピアはグレースフル リスタートをサポートします。

追加 BGP パス機能により、特定のプレフィックスにアドバタイズされるパス数が再起動の前後で同じ場合、パス ID の選択は古いパスの最終状態および削除を保証します。いくつかのパスが指定されたプレフィックスにアドバタイズされる場合、古いパスがグレースフル リスタート ヘルパー ピアに発生する可能性があります。

メモリ不足の処理

BGP は、次の条件でメモリ不足に対処します。

- マイナー アラート：BGP は新しい eBGP ピアを確立しません。BGP は新しい iBGP ピアおよび連合ピアの確立は続行します。ピアは存続しますが、リセット ピアは再確立されません。
- 重大アラート：BGP は、メモリ アラートがマイナーになるまで、選択した確立済み eBGP ピアを 2 分おきにシャット ダウンします。eBGP ピアごとに、受信したパスの合計数と最適パスとして選択されたパスの数の比率が計算されます。比率が最高のピアが、メモリ使用状況を削減するためのシャット ダウン対象として選択されます。オシレーションを回避するために、シャットダウンされた eBGP ピアを復帰する前にその eBGP ピアをクリアする必要があります。

（注） 重要な eBGP ピアをこの選択プロセスから除外できます。

- クリティカル アラート：BGP は確立されたすべてのピアを正常にシャット ダウンします。シャットダウンされた eBGP ピアを復帰する前にその eBGP ピアをクリアする必要があります。

メモリ不足状態によるシャットダウンから BGP ピアを除外する方法の詳細については、「BGP のチューニング」を参照してください。

仮想化のサポート

1 個の BGP インスタンスを設定できます。BGP は、仮想ルーティングおよび転送（VRF）インスタンスをサポートします。

拡張 BGP の前提条件

拡張 BGP の前提条件は次のとおりです。

- BGP を有効にする必要があります（「BGP の有効化」の項を参照）。
- システムに有効なルータ ID を設定しておく必要があります。
- Regional Internet Registry（RIR）によって割り当てられたか、またはローカル管理の AS 番号を取得しておく必要があります。
- ネイバー関係を作成しようとするピアに到達可能でなければなりません（Interior Gateway Protocol（IGP）、スタティック ルート、直接接続など）。
- BGP セッションを確立するネイバー環境で、アドレス ファミリを明示的に設定する必要があります。

拡張 BGP に関する注意事項と制限事項

拡張 BGP 設定時の注意事項および制約事項は、次のとおりです。

- Cisco NX-OS リリース 9.3(5) 以降、コマンドの動作が変更された 3 つのシナリオがあります。
  -
    ```
    Router bgp 1
        Template peer abc
            Ttl-security hops 30
        Neighbor 1.2.3.4
            Inherit peer abc
    ```

    後で **ebgp-multihop 20** コマンドを入力すると、**ttl-security hops 30** コマンドが存在するため、設定はブロックされます。Cisco NX-OS リリース 9.3(5) 以降、設定はブロックされなくなりました。ただし、**ttl-security hops** コマンドが優先され、有効な機能になります。

- ```
  Router bgp 1
      Template peer abc
          Ebgp-multihops 20
      Neighbor 1.2.3.4
          Inherit peer abc
  ```

  後で **ttl-security hops 30** コマンドを入力すると、**ebgp-multihop 20** コマンドが存在するため、設定はブロックされます。Cisco NX-OS リリース 9.3(5) 以降、設定はブロックされなくなりました。ただし、ここでも **ttl-security hops** コマンドが優先され、有効な機能なります。

- ```
  Router bgp 1
      Template peer abc
          Remote-as 1
      Neighbor 1.2.3.4
          Inherit peer abc
  ```

  後で **ttl-security hops 30** または **ebgp-multihop 20** コマンドを入力すると、ブロックされます。Cisco NX-OS リリース 9.3(5) 以降、設定はブロックされなくなりました。ただし、ピアが iBGP ピアになる **remote-as** コマンドが優先されるため、これらの機能はオフになります。

- プレフィックス ピアリングは、パッシブ TCP モードでのみ動作します。ピア アドレスがプレフィックス内にある場合、リモート ピアからの着信接続を受け入れます。
- Cisco NX-OS 9.3(5) 以降、vPC ピアへの TTL 値が 1 のパケットは、転送されるハードウェアです。
- **advertise-maps** コマンドを複数回設定することはサポートされていません。
- プレフィックス リスト内の名前は、大文字と小文字が区別されません。一意の名前を使用することを推奨します。大文字と小文字を変更して同じ名前を使用しないでください。たとえば、CTCPrimaryNetworks と CtcPrimaryNetworks は 2 つの異なるエントリではありません。
- ダイナミック AS 番号プレフィックス ピア設定は、BGP テンプレートから継承した個々の AS 番号の設定よりも優先します。
- AS 連合でプレフィックス ピアにダイナミック AS 番号を設定した場合、BGP はローカル連合の AS 番号のみでセッションを確立します。
- ダイナミック AS 番号プレフィックス ピアで作成された BGP セッションは、設定済みの eBGP マルチホップ存続可能時間（TTL）値や直接接続ピアに対するディセーブル済みのチェックを無視します。
- ルータ ID の自動変更およびセッション フラップを避けるために、BGP 用のルータ ID を設定します。
- ピアごとに最大プレフィックス設定オプションを使用し、受信するルート数および使用するシステム リソース数を制限してください。
- update-source を設定し、eBGP マルチホップ セッションでセッションを確立します。
- 再配布を設定する場合は、BGP ルート マップを指定します。

- VRF 内で BGP ルータ ID を設定します。
- キープアライブおよびホールドタイマーの値を小さくすると、ネットワークでセッションフラップが発生する可能性があります。
- BGP をIGP に再配布するとき、iBGP も再配布されます。この動作を無効にするには、ルートマップに追加 deny 文を挿入します。
- BGP へのエッジ サービス ゲートウェイ（ESG）ルートインジェクトは、重み 0 に割り当てられます。
- iBGP の単一ホップ ピアに対して BFD を有効にするには、物理インターフェイスの **update-source** オプションを設定します。
- Cisco NX-OS リリース 9.3(3) 以降では、BGP の BFD は BGP IPv4 と IPv6 のプレフィックス ピアでサポートされます。
- VLAN には、次の注意事項および制約事項が **remove-private-as** コマンドに適用されます。
  - これは、eBGP ピアにだけ適用されます。
  - これは、パブリック AS のみのルータのみに適用されます。この制約事項を回避するには、ネイバー単位で **neighbor local-as** コマンドを適用し、ローカル AS 番号をパブリック AS 番号として指定することです。
  - ネイバー コンフィギュレーション モードだけで設定可能となり、ネイバー アドレス ファミリ モードでは設定できません。
  - AS パスにプライベートとパブリック AS 番号を含める場合、プライベート AS 番号は削除されません。
  - AS パスに eBGP ネイバーの AS 番号が含まれている場合、プライベート AS 番号は削除されません。
  - その AS パス内のすべての AS 番号がプライベート AS 番号範囲に属する場合のみ、プライベート AS 番号は削除されます。ピアの AS 番号または非プライベート AS 番号が AS パス セグメントに存在する場合、プライベート AS 番号は削除されません。
- BGP テーブル内のネクストホップの一致ルートに AS-Path がない場合、または BGP テーブル内のネクストホップに一致するルートがない場合、traceroute 出力に、ネクストホップとともに AS 番号が出力されることはありません。
- **aggregate-address** を使用する場合 コマンドを使用して集約アドレスを設定し、**suppress-fib-pending** コマンドを使用して BGP ルートを抑制するコマンドを使用する場合、集約のロスレス トラフィックを BGP またはシステム トリガーで保証できません。
- スイッチでFIB抑制をイネーブルにし、ルートプログラミングがハードウェアで失敗すると、BGPはハードウェアでローカルにプログラミングされていないルートをアドバタイズします。
- ネイバー、テンプレート ピア、テンプレート ピアセッション、またはテンプレート ピアポリシー コンフィギュレーション モードでコマンドを無効にした場合（ **inherit peer** または **inherit peer-session** コマンドが存在する場合）、 **default** キーワードを使用してコマンドをデ

フォルトの状態に戻す必要があります。たとえば、実行コンフィギュレーションから **default update-source loopback 0** コマンドを無効にするには、 **update-source loopback 0** コマンドを入力する必要があります。

- route-reflector クライアントに next-hop-self が設定されている場合、ルート リフレクタは自身をネクスト ホップとしてクライアントにルートをアドバタイズします。
- 重み付き ECMP に次の注意事項および制約事項が適用されます。
  - 重み付き ECMP 機能は、IPv4 アドレス ファミリでのみサポートされます。
  - BGPは、draft-ietf-idr-link-bandwidth-06.txtで定義されているリンク帯域幅EXTCOMMを使用して、重み付けECMP機能を実装します。
  - BGP は、eBGP ピアと iBGP ピアの両方から受け入れることができます。
- IPv4 および IPv6 アドレス ファミリの IPv6 リンクローカルを介した BGP インターフェイス ピアリングには、次の注意事項と制限事項が適用されます。
  - この機能は、複数のインターフェイス間で同じリンクローカル アドレスを設定することをサポートしていません。
  - IP6 リンクローカル静的 IPv6 アドレスを使用して BGP インターフェイス ピアリングを構成する場合は、BGP が機能するようにサブネットがピアと一致していることを確認します。
  - この機能は、論理インターフェイス（ループバック）ではサポートされていません。イーサネット インターフェイス、ポートチャネル インターフェイス、サブインターフェイス、およびブレークアウト インターフェイスのみがサポートされます。
  - Cisco NX-OS リリース 9.3(6) 以降では、VLAN インターフェイスがサポートされます。
  - この機能は、リンクローカルアドレスを持つIPv6対応インターフェイスでのみサポートされます。
  - この機能は、設定されたプレフィックス ピアとインターフェイスのリモート ピアが同じ場合はサポートされません。
  - 次のコマンドはネイバー インターフェイス コンフィギュレーション モードではサポートされていません。
    - **disable-connected-check**
    - **maximum-peers**
    - **update-source**
    - **ebgp-multihop**
  - BFD マルチホップおよび次のコマンドは、IPv4 および IPv6 アドレス ファミリの IPv6 リンクローカルを介した BGP インターフェイス ピアリングではサポートされません。
    - **bfd-multihop**

- **bfd multihop interval**
- **bfd multihop authentication**

- BGPでは、ルート アドバタイズメントのコンバージェンス時間が短縮されます。ルート アドバタイズメント（RA）リンクレベル プロトコルの検出を高速化するには、IPv4 および IPv6 アドレス ファミリの IPv6 リンクローカル経由 BGP インターフェイス ピアリングを使用する各 IPv6 対応インターフェイスで次のコマンドを入力します。

```
interface Ethernet port/slot
ipv6 nd ra-interval 4 min 3
ipv6 nd ra-lifetime 10
```

- リンクローカルでBGPネイバーを設定する場合は、TCAM「in-sup」を512から768にカスタマイズする必要があります。ただし、デフォルトの切り分けで十分な Cisco Nexus 34XX-S プラットフォームは除きます。
- **[maximum-paths eibgp]** コマンドは、MPLS 環境でのみサポートされています。
- ルートマップ削除機能は、BGP に関連付けられたルートマップ全体の削除をブロックするメカニズムを追加します。ルート マップの削除がブロックされても、ルート マップ ステートメントへの変更は引き続き許可されます。
- ルート マップに複数のシーケンスがある場合、少なくとも 1 つのシーケンスが使用可能になるまで、ユーザーはルート マップ シーケンスを削除できます。
- ユーザーは、クライアントからのルート マップの前方参照ケースを持つことができます。ただし、ルートマップが作成されて関連付けられると、ルートマップの削除はブロックされます。
- ブロック削除機能は、ノブを使用して動的に構成できます。
- ルート マップへの BGP アソシエーションを削除すること、および単一のトランザクション ペイロードでルート マップ自体を削除することは許可されています。
- ルート マップに BGP アソシエーションを追加することが許可されており、ルート マップの削除に対してエラーをスローする必要があります。
- 以下は、デュアル ステージに関連する動作のリストです。
  - ノブと削除が同時に発生した場合、デュアル ステージは検証し、コミットせずにエラーをスローする必要があります。
  - ノブはすでに存在し、ルートマップ削除がデュアル ステージで発生する場合、エラーをスローする必要があります。
  - ルートマップと CLI ノブが異なる順序のシングル コミットである場合、エラーをスローする必要があります。
  - ノブが有効になっておらず、ルート マップの削除がデュアル ステージで発生した場合は、正常に実行する必要があります。

  - 1回のベリファイで、「cliノブが無効かつルートマップの削除」が実行された場合、ルートマップの削除が許可されます。

- BGP テンプレートで使用されるルート マップがいずれの BGP ネイバーにも継承されない場合、ルート マップ全体の削除は引き続きブロックされます。
- BGP によって所有されているが、bgpInst の一部ではない、vrf コンテキストの下にいくつかのコマンドがあります。
- Cloudscale IPv6 リンクローカル BGP のサポートには、512 を超える ing-sup TCAM リージョンを切り分ける必要があります (これを有効にするには、リロードが必要です)。
- VPN アドレス ファミリ (L3VPN および EVPN) がサポートされていないため、同盟ピアから受信したルートは VPN アドレス ファミリでアドバタイズされません。
- Cisco NX-OS リリース 10.3(1)F 以降、BGP は Cisco Nexus 9808 スイッチでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、BGP は Cisco Nexus 9804 スイッチでサポートされます。
- Cisco NX-OS リリース 10.3(1)F 以降、VXLAN EVPN は、Cisco Nexus 9808 スイッチで、トランジットとしてのみサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、VXLAN EVPN は、Cisco Nexus 9804 スイッチで、トランジットとしてのみサポートされます。
- Cisco NX-OS リリース 10.3(3)F 以降、BGP パスワードのタイプ 6 暗号化は、次の制限付きで Cisco NX-OS スイッチでサポートされます。
  - タイプ 6 暗号化が構成されている場合、既存のタイプ 6 暗号化パスワードをタイプ 0/タイプ 3/タイプ 7 パスワードに変更することはできません。
  - タイプ 6 暗号化がサポートされていない古いイメージでコールド リブートによってシステムをダウングレードする場合は、タイプ 6 構成を削除して、それからコールド リブートを実行してください。そうしないと、構成が失われ、ネイバーの構成がなくなります。
  - プライマリ キーの設定は、スイッチに対してローカルです。あるスイッチからタイプ 6 に構成された実行データを取得し、別のプライマリ キーが構成されている別のスイッチに適用すると、新しいスイッチでの復号化は失敗します。
  - ISSU 中に、古いイメージ（タイプ 0/タイプ 3/タイプ 7 暗号化キーが構成に存在する）から新しいイメージ（タイプ 6 暗号化がサポートされている）に移行する場合、BGP は既存**encryption re-encrypt obfuscated** のコマンドを使用して再暗号化が適用されるまで、または適用されない限り、タイプ 6 の暗号化に既存のキーを変換しません。
  - BGP タイプ 6 パスワードは、非 DME プラットフォームではサポートされません。
  - ネイバーまたはテンプレートのパスワードをプログラム（RESTCONF、NETCONF など）で設定する場合は、パスワードのタイプとパスワードを指定することを強くお勧めします。プログラム コールでいずれかのプロパティが欠落している場合、BGP は欠落しているプロパティのすでに使用可能な（またはデフォルトの）値を使用して、ネイバーまたはテンプレートのパスワードを構成します。

ユーザーがプロパティを指定せずに構成する必要がある場合、ユーザーは両方のピアルータで同じ手順を実行する必要があります。

- Cisco NX-OS リリース 10.4(1)F 以降、BGP は、Cisco Nexus 9808 および 9804 スイッチを搭載した Cisco Nexus X98900CD-A および X9836DM-A ライン カードでサポートされます。

デフォルト設定

高度な BGP パラメータのデフォルト設定値を表に示します。

パラメータ	デフォルト
BGP 機能	ディセーブル
BGP の追加パス	ディセーブル
キープアライブインターバル	60 秒
ホールド タイマー	180 秒
ダイナミック機能	有効（Enabled）

高度な BGP の設定

インターフェイスでの IP 転送の有効化

RFC 5549 を使用するには、少なくとも 1 つの IPv4 アドレスを設定する必要があります。IPv4 アドレスを設定しない場合は、RFC 5549 を使用するように IP転送機能を有効にする必要があります。

手順の概要

1. **configure terminal**
2. **interface** *type slot/port*
3. **ip forward**
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**interface** *type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**ip forward** 例： `switch(config-if)# ip forward`	インターフェイスに IP アドレスが設定されていない場合でも、そのインターフェイスで IPv4 トラフィックを許可します。
Step 4	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

BGP セッション テンプレートの設定

BGP セッション テンプレートを使用すると、類似した設定が必要な複数の BGP ピアで、BGP の設定を簡素化できます。BGPテンプレートによって、共通のコンフィギュレーションブロックを再利用できます。先に BGP テンプレートを設定し、BGP ピアにテンプレートを適用します。

BGPセッションテンプレートでは、継承、パスワード、タイマー、セキュリティなどのセッション属性を設定できます。

peer-sessionテンプレートは、別のpeer-sessionテンプレートからの継承が可能です。第3のテンプレートから継承するように第2テンプレートを設定できます。さらに最初のテンプレートもこの第3のテンプレートから継承させることができます。この間接継承を続けることができるpeer-sessionテンプレートの数は、最大7つです。

ネイバーに設定した属性は、ネイバーが BGP テンプレートから継承した属性よりも優先されます。

始める前に

BGP を有効にする必要があります（「BGP の有効化」の項を参照）。

（注）

- テンプレートを編集するときには、ピアまたはテンプレートのレベルで **no** 形式のコマンドを使用すると、テンプレートの設定を明示的に上書きできます。属性をデフォルトの状態にリセットするには、default 形式のコマンドを使用する必要があります。
- BGP ピア テンプレートを使用する場合、テンプレート内で使用されるコマンドをチェックして、そのコマンドが iBGP / eBGP ピアに適用されるかどうかを確認することはありません。たとえば、テンプレートを作成し、テンプレート内に「Remove-private-as」コマンドを追加し、このテンプレートを iBGP ピアに割り当てた場合、このコマンド「Remove-private-as」は適用されないというエラーは出力されません。 iBGP ピア。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **template peer-session** *template-name*
4. （任意） **password** *number password*
5. （任意） **timers** *keepalive hold*
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **inherit peer-session** *template-name*
9. （任意） **description** *text*
10. （任意） **show bgp peer-session** *template-name*
11. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに自律システム番号を割り当てます。

Step 3	**template peer-session** *template-name* 例：	peer-session テンプレート コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	`switch(config-router)# template peer-session BaseSession` `switch(config-router-stmp)#`	
Step 4	（任意） **password** *number password* 例： `switch(config-router-stmp)# password 0 test`	ネイバーにクリア テキストのパスワード「test」を追加します。パスワードは 3DES（タイプ 3 暗号形式）で保存および表示されます。
Step 5	（任意） **timers** *keepalive hold* 例： `switch(config-router-stmp)# timers 30 90`	peer-session テンプレートに BGP キープアライブおよびホールドタイマー値を追加します。 デフォルトのキープアライブ インターバルは 60 です。デフォルトのホールド タイムは 180 です。

Step 6	**exit**
	例：
	`switch(config-router-stmp)# exit`
	`switch(config-router)#`
		peer-session テンプレート コンフィギュレーション モードを終了します。

Step 7	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65535`
	`switch(config-router-neighbor)#`
		BGP ルーティング用のネイバー コンフィギュレーション モードを開始し、ネイバー IP アドレスを設定します。

Step 8	**inherit peer-session** *template-name*
	例：
	`switch(config-router-neighbor)# inherit peer-session BaseSession`
	`switch(config-router-neighbor)#`
		ピアに peer-session テンプレートを適用します。

Step 9	（任意） **description** *text*
	例：
	`switch(config-router-neighbor)# description Peer Router A`
	`switch(config-router-neighbor)#`
		ネイバーの説明を追加します。

Step 10	（任意） **show bgp peer-session** *template-name* 例： `switch(config-router-neighbor)# show bgp peer-session BaseSession`	peer-policy テンプレートを表示します。
Step 11	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor)# copy running-config startup-config`	この設定変更を保存します。 **show bgp neighbor** コマンドを使用し、 コマンドを実行して、適用されたテンプレートを確認します。

例

BGP peer-session テンプレートを設定して、BGP ピアに適用する例を示します。

```
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-af)# copy running-config startup-config
```

BGP peer-policy テンプレートの設定

peer-policyテンプレートを設定すると、特定のアドレス ファミリに対応する属性を定義できます。各 peer-policy テンプレートにプリファレンスを割り当て、指定した順序でテンプレートが継承されるようにします。ネイバー アドレス ファミリでは最大 5 つの peer-policy テンプレートを使用できます。

Cisco NX-OS は、プリファレンス値を使用して、アドレス ファミリの複数のピア ポリシーを評価します。プリファレンス値が最小のものが最初に評価されます。ネイバーに設定した属性は、ネイバーが BGP テンプレートから継承した属性よりも優先されます。

peer-policy テンプレートでは、AS-path フィルタ リスト、プレフィックス リスト、ルート リフレクション、ソフト再構成など、アドレス ファミリ固有の属性を設定できます。

（注） **show bgp neighbor** コマンドを使用し、 コマンドを実行して、適用されたテンプレートを確認します。テンプレートで使用できる全コマンドの詳細については、『*Cisco Nexus 9000 Series NX-OS Unicast Routing Command Reference*』を参照してください。

始める前に

BGP を有効にする必要があります（「BGP の有効化」の項を参照）。

（注） テンプレートを編集するときには、ピアまたはテンプレートのレベルで **no** 形式のコマンドを使用すると、テンプレートの設定を明示的に上書きできます。属性をデフォルトの状態にリセットするには、default 形式のコマンドを使用する必要があります。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*

3. **template peer-session** *template-name*
4. （任意） **advertise-active-only**
5. （任意） **maximum-prefix** *number*
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **address-family** {**ipv4** | **ipv6**} {**multicast** | **unicast**}
9. **inherit peer-policy** *template-name preference*
10. （任意） **show bgp peer-policy** *template-name*
11. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例： `switch# configure terminal`	コンフィギュレーション モードに入ります。

Step 2	**router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに自律システム番号を割り当てます。

Step 3	**template peer-session** *template-name*
	例：
	`switch(config-router)# template peer-policy BasePolicy`
	`switch(config-router-ptmp)#`
		peer-policy テンプレートを作成します。

Step 4	（任意） **advertise-active-only** 例： `switch(config-router-ptmp)# advertise-active-only`	アクティブ ルートのみをピアにアドバタイズします。
Step 5	（任意） **maximum-prefix** *number* 例： `switch(config-router-ptmp)# maximum-prefix 20`	このピアに認めるプレフィックスの最大数を設定します。

Step 6	**exit**
	例：
	`switch(config-router-ptmp)# exit`
	`switch(config-router)#`
		peer-policy テンプレート コンフィギュレーション モードを終了します。

	コマンドまたはアクション	目的

Step 7	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65535`
	`switch(config-router-neighbor)#`
		BGP ルーティング用のネイバー コンフィギュレーション モードを開始し、ネイバー IP アドレスを設定します。

Step 8	**address-family** {**ipv4** | **ipv6**} {**multicast** | **unicast**}
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		指定のアドレス ファミリに対しグローバル アドレス ファミリ設定モードを開始します。

Step 9	**inherit peer-policy** *template-name preference* 例： `switch(config-router-neighbor-af)# inherit peer-policy BasePolicy 1`	ピア アドレス ファミリ設定に peer-policy テンプレートを適用し、このピア ポリシーのプリファレンス値を割り当てます。
Step 10	（任意） **show bgp peer-policy** *template-name* 例： `switch(config-router-neighbor-af)# show bgp peer-policy BasePolicy`	peer-policy テンプレートを表示します。
Step 11	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor-af)# copy running-config startup-config`	この設定変更を保存します。 **show bgp neighbor** コマンドを使用し、 コマンドを実行して、適用されたテンプレートを確認します。

例

BGP peer-policy テンプレートを設定して、BGP ピアに適用する例を示します。

```
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
```

BGP peer テンプレートの設定

BGP peer テンプレートを設定すると、1 つの再利用可能なコンフィギュレーション ブロックで、セッション属性とポリシー属性を結合することができます。peer テンプレートも、peer-session または peer-policy テンプレートを継承できます。ネイバーに設定した属性は、ネイバーが BGP テン

プレートから継承した属性よりも優先されます。ネイバーに設定できる peer テンプレートは 1 つだけですが、peer テンプレートは peer-session および peer-policy テンプレートを継承できます。

peer テンプレートは、eBGP マルチホップ TTL、最大プレフィックス数、ネクストホップ セルフ、タイマーなど、セッション属性およびアドレス ファミリ属性をサポートします。

始める前に

BGP を有効にする必要があります（「BGP の有効化」の項を参照）。

（注） テンプレートを編集するときには、ピアまたはテンプレートのレベルで **no** 形式のコマンドを使用すると、テンプレートの設定を明示的に上書きできます。属性をデフォルトの状態にリセットするには、default 形式のコマンドを使用する必要があります。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **template peer** *template-name*
4. （任意） **inherit peer-session** *template-name*
5. （任意） **address-family** {**ipv4**|**ipv6**} {**multicast**|**unicast**}
6. （任意） **inherit peer-policy** *template-name*
7. **exit**
8. （任意） **timers** *keepalive hold*
9. **exit**
10. **neighbor** *ip-address* **remote-as** *as-number*
11. **inherit peer** *template-name*
12. （任意） **timers** *keepalive hold*
13. （任意） **show bgp peer-template** *template-name*
14. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例： `switch# configure terminal`	グローバル コンフィギュレーション モードを開始します。
Step 2	**router bgp** *autonomous-system-number* 例： `switch(config)# router bgp 65535`	BGP モードを開始し、ローカル BGP スピーカに自律システム番号を割り当てます。

	コマンドまたはアクション	目的
Step 3	**template peer** *template-name* 例： `switch(config-router)# template peer BasePeer`	peer テンプレート コンフィギュレーション モードを開始します。
Step 4	（任意） **inherit peer-session** *template-name* 例： `switch(config-router-neighbor)# inherit peer-session BaseSession`	ピア テンプレートに peer-session テンプレートを適用します。

Step 5	（任意） **address-family** {**ipv4**|**ipv6**} {**multicast**|**unicast**}
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)`
		指定のアドレス ファミリに対しグローバル アドレス ファミリ コンフィギュレーション モードを設定します。

Step 6	（任意） **inherit peer-policy** *template-name* 例： `switch(config-router-neighbor-af)# inherit peer-policy BasePolicy 1`	ネイバー アドレス ファミリ設定に peer-policy テンプレートを適用します。
Step 7	**exit** 例： `switch(config-router-neighbor-af)# exit`	BGP ネイバー アドレス ファミリ コンフィギュレーション モードを終了します。
Step 8	（任意） **timers** *keepalive hold* 例： `switch(config-router-neighbor)# timers 45 100`	ピアに BGP タイマー値を追加します。 これらの値によって、peer-session テンプレート、BaseSession のタイマー値が上書きされます。
Step 9	**exit** 例： `switch(config-router-neighbor)# exit`	BGP ネイバー コンフィギュレーション モードを終了します。

Step 10	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65535`
	`switch(config-router-neighbor)#`
		BGP ルーティング用のネイバー設定モードを開始し、ネイバー IP アドレスを設定します。

Step 11	**inherit peer** *template-name* 例： `switch(config-router-neighbor)# inherit peer BasePeer`	peer テンプレートを継承します。

	コマンドまたはアクション	目的
Step 12	（任意） **timers** *keepalive hold* 例： `switch(config-router-neighbor)# timers 60 120`	このネイバーに BGP タイマー値を追加します。 これらの値によって、peer テンプレートおよび peer-session テンプレートのタイマー値が上書きされます。
Step 13	（任意） **show bgp peer-template** *template-name* 例： `switch(config-router-neighbor)# show bgp peer-template BasePeer`	peer テンプレートを表示します。
Step 14	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor)# copy running-config startup-config`	この設定変更を保存します。 **show bgp neighbor** コマンドを使用し、 コマンドを実行して、適用されたテンプレートを確認します。

例

BGP peer テンプレートを設定して、BGP ピアに適用する例を示します。

```
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
```

プレフィックス ピアリングの設定

BGP では、IPv4 と IPv6 の両方のプレフィックスを使用してピア セットを定義できます。この機能を使用すると、各ネイバーを設定に追加する必要がありません。

プレフィックス ピアリングを定義する場合は、プレフィックスとともにリモート AS 番号を指定する必要があります。プレフィックス ピアリングが設定されている許容最大ピア数を超えない場合、BGP はプレフィックスおよび自律システムから接続するピアを受け付けます。

プレフィックス ピアリングに含まれている BGP ピアが切断されると、Cisco NX-OS は定義されているプレフィックス ピアタイムアウト値まで、ピア構造を維持します。この場合、そのプレフィックス ピアリングのすべてのスロットを他のピアが使い果たした結果、ブロックされるという危険性を伴わずに、確立されたピアのリセットまたは再接続が可能になります。

手順の概要

1. **timers prefix-peer-timeout** *value*
2. **maximum-peers** *value*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**timers prefix-peer-timeout** *value* 例： `switch(config-router-neighbor)# timers prefix-peer-timeout 120`	ルータ コンフィギュレーション モードで BGP プレフィックス ピアリングのタイムアウト値を設定します。有効な範囲は 0 ～ 1200 秒です。デフォルト値は 30 です。 （注） プレフィックス ピアの場合は、プレフィックス ピア タイムアウトを、設定されたグレースフル リスタート タイマーよりも大きく設定します。プレフィックス ピア タイムアウトがグレースフル リスタートタイマーよりも大きければ、ピアのルートは再起動中に保持されます。プレフィックス ピア タイムアウトがグレースフル リスタート タイマーよりも小さいと、ピアのルートはプレフィックス ピア タイムアウトによって消去されます。これは、再起動が完了する前に発生する可能性があります。
Step 2	**maximum-peers** *value* 例： `switch(config-router-neighbor)# maximum-peers 120`	ネイバー設定モードのこのプレフィックス ピアリングの最大ピア数を設定します。範囲は 1 ～ 1000 です。

例

最大 10 のピアを受け付けるプレフィックス ピアリングの設定例を示します。

```
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)#
```

show bgp ipv4 unicast neighbors コマンドを使用し、 すると、所定のプレフィックス ピアリングの設定の詳細とともに、現在受け付けられているインスタンスのリスト、アクティブ ピア数、最大同時ピア数、および受け付けたピアの合計数を表示できます。

IPv4 および IPv6 アドレス ファミリ向け IPv6 リンク ローカル経由の BGP インターフェイス ピアリングの設定

アンナンバード インターフェイスを使用した自動 BGP ネイバー探索のために、IPv4 および IPv6 アドレス ファミリの IPv6 リンクローカルを経由して、BGP インターフェイス ピアリングを設定できます。これにより、インターフェイス名を（インターフェイススコープのアドレスではなく）BGP ピアとして使用する BGP セッションを設定できます。この機能は、ICMPv6 ネイバー探索（ND）のルート アドバタイズメント（RA）を使用して自動ネイバー探索を行い、RFC 5549 を使用して IPv6 ネクスト ホップで IPv4 ルートを送信します。

始める前に

BGP をイネーブルにする必要があります（「BGP のイネーブル化」の項を参照）。

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例： `switch# configure terminal`	コンフィギュレーション モードに入ります。

Step 2	**router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに自律システム番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

Step 3	**neighbor** *interface-name* **remote-as** {*as-number* | **route-map** *map-name*}
	例：
	`switch(config-router)# neighbor Ethernet1/1 remote-as 65535`
	`switch(config-router-neighbor)#`
		BGP ルーティングのためにルータをネイバー設定モードにして、インターフェイスを BGP ピア用に設定します。
		（注）
		指定できるのは、イーサネットインターフェイス、ポートチャネル インターフェイス、サブインターフェイス、およびブレークアウトインターフェイスだけです。
		Cisco NX-OS リリース 9.3(6) 以降では、ルート マップを指定でき、AS リストを含められるルート マップを指定できます。ダイナミック AS 番号の使用の詳細については、プレフィックス ピアおよびインターフェイス ピアのダイナミック AS 番号（321 ページ） を参照してください。

	コマンドまたはアクション	目的
		設定を複数のインターフェイスに適用する必要がある場合、*interface-name* は範囲にすることができます。
Step 4	**inherit peer** *template-name* 例： `switch(config-router-neighbor)# inherit peer PEER`	peer テンプレートを継承します。
Step 5	**address-family** {**ipv4** | **ipv6**} **unicast** 例： `switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#`	指定のアドレス ファミリに対しグローバルアドレス ファミリ設定モードを開始します。

Step 6	（任意） **show bgp** {**ipv4** | **ipv6**} **unicast neighbors** *interface*
	例：
	`switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors e1/25`
	例：
	`switch(config-router-neighbor-af)# show bgp ipv6 unicast neighbors 3FFE:700:20:1::11`
		BGP ピアに関する情報を表示します。

Step 7	（任意） **show ip bgp neighbors** *interface-name* 例： `switch(config-router-neighbor-af)# show ip bgp neighbors Ethernet1/1`	BGP ピアとして使用されるインターフェイスを表示します。
Step 8	（任意） **show ipv6 routers** [**interface** *interface*] 例： `switch(config-router-neighbor-af)# show ipv6 routers interface Ethernet1/1`	IPv6 ICMP ルータ アドバタイズメントによって学習されたリモート IPv6 ルータのリンク ローカル アドレスを表示します。
Step 9	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、IPv4 および IPv6 アドレス ファミリの IPv6 リンクローカル経由で、BGP インターフェイス ピアリングを設定する例を示します。

リーフ 1 の iBGP インターフェイス ピアリング設定：

```
switch# configure terminal
switch(config)# router bgp 65000
switch(config-router)# neighbor Ethernet1/1 remote-as 65000
switch(config-router-neighbor)# inherit peer PEER
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor)# address-family ipv6 unicast
switch(config-router-neighbor-af)# copy running-config startup-config
```

次に、IPv4 および IPv6 アドレス ファミリの IPv6 リンクローカル経由での、BGP インターフェイス ピアリングのサンプル出力例を示します。

```
switch(config-router-neighbor)# show bgp ipv4 unicast neighbors e1/15.1
BGP neighbor is fe80::2, remote AS 100, ibgp link, Peer index 4
Peer is an instance of interface peering Ethernet1/15.1
BGP version 4, remote router ID 5.5.5.5
Neighbor previous state = OpenConfirm
BGP state = Established, up for 2d16h
Neighbor vrf: default
Peer is directly attached, interface Ethernet1/15.1
Last read 00:00:54, hold time = 180, keepalive interval is 60 seconds
Last written 00:00:08, keepalive timer expiry due 00:00:51
Received 3869 messages, 0 notifications, 0 bytes in queue
Sent 3871 messages, 0 notifications, 0(0) bytes in queue
Enhanced error processing: On
0 discarded attributes
Connections established 2, dropped 1
Last reset by peer 2d16h, due to session closed
Last error length received: 0
Reset error value received 0
Reset error received major: 104 minor: 0
Notification data received:
Last reset by us never, due to No error
Last error length sent: 0
Reset error value sent: 0
Reset error sent major: 0 minor: 0
--More--
```

インターフェイス コンフィギュレーション：

次のいずれかのコマンドを使用して、対応するインターフェイスで IPv6 を有効にする必要があります。

- **ipv6 address** *ipv6-address*
- **ipv6 address use-link-local-only**
- **ipv6 link-local** *link-local-address*

```
switch# configure terminal
switch(config)# interface Ethernet1/1
switch(config-if)# ipv6 address use-link-local-only
```

（注） インターフェイスで IPv4 アドレスが設定されていない場合は、**ip forward** コマンドをインターフェイスで設定して IPv4 転送を有効にする必要があります。

（注） IPv6 ND タイマーを調整して、ネイバー探索を高速化し、BGP のルートコンバージェンスを高速化できます。

```
switch(config-if)# ipv6 nd ra-interval 4 min 3
switch(config-if)# ipv6 nd ra-lifetime 10
```

（注） Cisco NX-OS リリース 9.3(6) 以降で、パラレル リンクを使用するカスタマーの導入では、インターフェイス モードで次のコマンドを追加する必要があります。

```
switch(config-if)# ipv6 link-local use-bia
```

このコマンドは、異なるインターフェイス間での IPv6 LLA を一意にします。

BGP 認証の設定

MD5 ダイジェストを使用してピアからのルート更新を認証するように、BGP を設定できます。

または、Cisco NX-OS リリース 10.4(2)F 以降では、TCP 認証オプション（TCP AO）を使用してピアからのルート アップデートを認証するように BGP を構成できます。

Cisco NX-OS リリース 10.3(3)F 以降では、BGP パスワードのタイプ 6 暗号化が Cisco NX-OS スイッチでサポートされています。次の暗号化タイプがサポートされています。

- AES ベースの暗号化
- 秘密の暗号化と復号には、プライマリキーと呼ばれる構成可能な暗号キーが使用されます。

MD5 ダイジェストまたは TCP AO を使用するように BGP を構成するには、ネイバー コンフィギュレーション モードで次のコマンドを使用します。

始める前に

- プライマリキーが Cisco NX-OS スイッチで **key config-key ascii** <*primary_key*> コマンドを使用して構成されていることを確認します。
- タイプ 6 暗号化を適切に機能させるには、Cisco NX-OS スイッチで **feature password encryption aes** が有効になっていることを確認します。
- BGP ネイバーセッション認証に TCP キーチェーン認証オプションを構成して使用するには、「TCP 認証オプションの構成」を参照してください。https://www.cisco.com/content/en/us/td/docs/dcn/nx-os/nexus9000/104x/configuration/security/cisco-nexus-9000-series-nx-os-security-configuration-guide-release-104x/chapter.html

手順の概要

1. **key config-key ascii** <*primary_key*>

2. **configure terminal**
3. **feature password encryption aes**
4. **router bgp** *AS* 番号
5. **template peer**テンプレート名
6. **password** {**0** | **3** | **7** | **6**} *string*
7. （任意） **encryption re-encrypt obfuscated**
8. （任意） **encryption delete type-6**
9. （任意） **ao** <*Keychain-name*> [**include-tcp-options**]

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**key config-key ascii** <*primary_key*> 例： `switch# key config-key ascii 0123456789012345`	プライマリキーを構成します。 （注） • このコマンドは、プライマリ キーが構成されていない場合にのみ入力します。 • プライマリ キーがすでに構成されている場合にこのコマンドを入力すると、実際には既存のプライマリ キー値が変更されます。新しい値に変更するには、プロンプトが表示されたら既存のプライマリ キー値を入力します。
Step 2	**configure terminal** 例： `switch# configure terminal`	グローバルコンフィギュレーションモードを開始します。
Step 3	**feature password encryption aes** 例： `switch(config)# feature password encryption aes`	AES パスワード暗号化を有効にします。
Step 4	**router bgp** *AS* 番号 例： `switch(config-router)# router bgp 1`	BGP ルータ モードを開始します。
Step 5	**template peer**テンプレート名 例： `switch(config-router-neighbor)# template peer abc`	BGP ネイバー モードを開始します。

	コマンドまたはアクション	目的
Step 6	**password** {**0** | **3** | **7** | **6**} *string* 例： `switch(config-router-neighbor)# password 6 JV5Ub2Vh7aVyC2c3YQc/ZlcnKR2PNjx/R6wrgWrHphwsQ0eP9rk8QDsc3XGTOA=`	MGP ネイバー セッションの MD5 パスワードを設定します。 （注） タイプ 0/タイプ 3/タイプ 7 を新しく構成する場合、プライマリ キーが構成されていて **feature password encryption aes** が有効になっている場合、タイプ 0/3/7 はタイプ 6 パスワードに自動的に暗号化されます。
Step 7	（任意） **encryption re-encrypt obfuscated** 例： `switch# encryption re-encrypt obfuscated`	既存のタイプ 0/タイプ 3/タイプ 7 パスワードをタイプ 6 パスワードに暗号化します。
Step 8	（任意） **encryption delete type-6** 例： `switch# encryption delete type-6`	タイプ 6 暗号化パスワードを削除します。
Step 9	（任意） **ao** <*Keychain-name*> [**include-tcp-options**]	パケットの MAC ダイジェストの計算中に TCP オプション ヘッダー（TCP AO オプション以外）を含めるかどうかを指定するオプションを構成します。

BGP セッションのリセット

BGP のルート ポリシーを変更した場合は、関連付けられた BGP ピア セッションをリセットする必要があります。BGP ピアがルート リフレッシュをサポートしない場合は、着信ポリシー変更に関するソフト再構成を設定できます。Cisco NX-OS は自動的に、セッションのソフト リセットを試みます。

ソフト再構成着信を設定するには、ネイバーアドレスファミリ設定モードで次のコマンドを使用します。

手順の概要

1. **soft-reconfiguration inbound**
2. （任意） **clear bgp** {**ipv4** | **ipv6** } {**unicast** | **multicast** *ip-address* **soft** {**in** | **out**}
3. **clear bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} *ip-address* **soft** (**in** | **out**)

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**soft-reconfiguration inbound** 例： `switch(config-router-neighbor-af)# soft-reconfiguration inbound`	着信BGPルートアップデートを格納するために、ソフト再構成をイネーブルにします。このコマンドによって、BGPネイバーセッションの自動ソフトクリアまたはリフレッシュが開始されます。
Step 2	（任意） **clear bgp** {**ipv4** | **ipv6** } {**unicast** | **multicast** *ip-address* **soft** {**in** | **out**} 例： `switch# clear bgp ip unicast 192.0.2.1 soft in`	TCP セッションを切断しないで、BGP セッションをリセットします。
Step 3	**clear bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} *ip-address* **soft** (**in** | **out**) 例： `switch# clear bgp ip unicast 192.0.2.1 soft in`	TCP セッションを切断しないで、BGP セッションをリセットします。

ネクストホップ アドレスの変更

次の方法で、ルート アドバタイズメントで使用するネクスト ホップ アドレスを変更できます。

- ネクストホップ計算をディセーブルにして、ローカルBGPスピーカアドレスをネクストホップ アドレスとして使用します。
- ネクストホップアドレスをサードパーティアドレスとして設定します。この機能は、元のネクスト ホップ アドレスがルートの送り先のピアと同じサブネット上にある場合に使用します。この機能を使用すると、フォワーディング時に余分なホップを節約できます。

ネクストホップアドレストラッキングを変更するには、アドレスファミリ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **next-hop-self**
2. **next-hop-third-party**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**next-hop-self** 例： `switch(config-router-neighbor-af)# next-hop-self`	ルートアップデートのネクストホップアドレスとして、ローカルBGPスピーカアドレスを使用します。このコマンドによって、BGPネイバーセッションの自動ソフト クリアまたはリフレッシュが開始されます。
Step 2	**next-hop-third-party** 例： `switch(config-router-neighbor-af)# next-hop-third-party`	ネクストホップアドレスをサードパーティアドレスとして設定します。このコマンドは、**next-hop-self** が設定されていないシングル ホップの EBGP ピアに使用します。 configured.

BGP ネクスト ホップ アドレス トラッキングの設定

BGP ネクスト ホップ アドレス トラッキングはデフォルトで有効であり、無効にすることができません。

BGP ネクスト ホップ トラッキングのパフォーマンスを向上するために、RIB チェック間の遅延インターバルを変更できます。

BGP ネクスト ホップ アドレス トラッキングを変更するには、アドレス ファミリ設定モードで次のコマンドを使用します。

手順の概要

1. **nexthop trigger-delay** {**critical** | **non-critical**} *milliseconds*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**nexthop trigger-delay** {**critical** | **non-critical**} *milliseconds* 例： `switch(config-router-af)# nexthop trigger-delay critical 5000`	クリティカルなネクスト ホップの到達可能性ルートおよび非クリティカルなルートについて、ネクストホップアドレストラッキングの遅延タイマーを指定します。指定できる範囲は 1 ～ 4294967295 ミリ秒です。クリティカル タイマーのデフォルトは 3000 です。非クリティカル タイマーのデフォルトは 10000 です。

ネクスト ホップ フィルタリングの設定

BGP ネクストホップ フィルタリングを使用すると、RIB でネクストホップ アドレスがチェックされるときにそのネクストホップアドレスの基盤となるルートがルートマップを経由します。ルートマップでそのルートが拒否されると、ネクストホップアドレスは到達不能として扱われます。

BGP は、ルート ポリシーによって拒否されたすべてのネクストホップを無効であるとマークし、無効なネクストホップ アドレスを使用するルートについてベスト パスを計算しません。

BGP ネクストホップ フィルタリングを設定するには、アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **nexthop route-map** *name*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**nexthop route-map** *name* 例: `switch(config-router-af)# nexthop route-map nextHopLimits`	BGP ネクストホップ ルートが一致するルート マップを指定します。63 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

デフォルト ルートによるネクストホップ解決の設定

BGP ネクストホップ解決では、IP デフォルト ルートを BGP ネクストホップ解決に使用するかどうかを指定できます。

BGP ネクストホップ解決を設定するには、ルータ設定モードで次のコマンドを使用します。

手順の概要

1. [**no**] **nexthop suppress-default-resolution**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	[**no**] **nexthop suppress-default-resolution** 例:	IP デフォルト ルートを介した BGP ネクストホップの解決を防止します。

	コマンドまたはアクション	目的
	`switch(config-router)# nexthop suppress-default-resolution`	このコマンドを有効にすると、以下のようになります。 • **show bgp process detail** コマンドの出力には、次の行が含まれます。 Use default route for nexthop Resolution：No • **show routing clients bgp** コマンドの出力には、次の行が含まれます。 Owned rnh will never resolve to 0.0.0.0/0

ネクストホップセルフによるリフレクト ルートの制御

NX-OS では、**next-hop-self** [all] 引数を使用して特定のピアに送信する際の iBGP ルートを制御できます。これらの引数を使用すると、ルートのリフレクトが実施されている場合でも、ルートのネクストホップを選択的に変更できます。

コマンド	目的
next-hop-self [all] 例： `switch(config-router-af)# next-hop-self all`	ルート アップデートのネクストホップ アドレスとして、ローカル BGP スピーカ アドレスを使用します。 all キーワードはオプションです。all を指定すると、すべてのルートが next-hop-self を使用するピアに送信されます。all を指定しなかった場合、リフレクトしたルートのネクストホップは変更されません。

セッションがダウンした場合のネクストホップ グループの縮小

セッションがダウンしたときに迅速な方法でECMPグループを縮小するようにBGPを設定できます。

この機能は、次の BGP パス障害イベントに適用されます。

- 1 つまたは複数のレイヤ 3 リンクの障害
- ライン カード障害
- BGP ネイバーの BFD 障害検出
- BGP ネイバーの管理上のシャットダウン（shutdown コマンドを使用）

最初の 2 つのイベント（レイヤ 3 リンク障害とラインカード障害）の迅速な処理はデフォルトでイネーブルになっており、イネーブルにするための設定コマンドは必要ありません。

最後の 2 つのイベントの迅速な処理を設定するには、ルータ設定モードで次のコマンドを使用します。

手順の概要

1. **neighbor-down fib-accelerate**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**neighbor-down fib-accelerate** 例： `switch(config-router)# neighbor-down fib-accelerate`	BGP セッションがダウンするたびに、すべてのネクストホップ グループ（ECMP グループと単一のネクストホップルート）から対応する次のネクストホップを取り消します。 （注） このコマンドは、IPv4 ルートと IPv6 ルートの両方に適用されます。

機能ネゴシエーションのディセーブル化

機能ネゴシエーションをディセーブルにすると、機能ネゴシエーションをサポートしない古い BGP ピアとの相互運用が可能です。

機能ネゴシエーションをディセーブルにするには、ネイバーコンフィギュレーションモードで次のコマンドを使用します。

手順の概要

1. **dont-capability-negotiate**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**dont-capability-negotiate** 例： `switch(config-router-neighbor)# dont-capability-negotiate`	機能ネゴシエーションをディセーブルにします。このコマンドの設定後、BGPセッションを手動でリセットする必要があります。

ポリシーのバッチ処理の無効化

プレフィックスに一意の属性がある BGP 展開では、BGP は、同じ BGP アップデートメッセージでバンドルする類似の属性を持つルートを識別しようとします。この追加の BGP 処理のオーバーヘッドを回避するには、バッチ処理をディセーブルにします。

固有のネクストホップを持つ多数のルートがある BGP 展開では、ポリシーバッチ処理を無効にすることを推奨します。

ポリシー バッチ処理を無効にするには、ルータ設定モードで次のコマンドを使用します。

手順の概要

1. **disable-policy-batching**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**disable-policy-batching** 例： `switch(config-router)# disable-policy-batching`	すべてのピアへのプレフィックスアドバタイズメントのバッチ評価をディセーブルにします。

BGP 追加パスの設定

BGP は、プレフィックスごとの複数パスの送受信と、このパスのアドバタイジングをサポートします。

追加パスの送受信機能のアドバタイズ

BGP ピア間の追加パスの送受信機能をアドバタイズするように BGP を設定できます。これを行うには、ネイバー アドレス ファミリ設定モードで次のコマンドを使用します。

手順の概要

1. [**no**] **capability additional-paths send** [**disable**]
2. [**no**] **capability additional-paths receive** [**disable**]
3. **show bgp neighbor**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	[**no**] **capability additional-paths send** [**disable**] 例： `switch(config-router-neighbor-af)# capability addtional-paths send`	BGP ピアに追加パスを送信する機能をアドバタイズします。**disable** オプションは、追加パス送信機能のアドバタイズをディセーブルにします。 このコマンドの **no** 形式を使用すると、追加パスの送信機能がディセーブルになります。
Step 2	[**no**] **capability additional-paths receive** [**disable**] 例： `switch(config-router-neighbor-af)# capability addtional-paths receive`	BGP ピアから追加パスを受信する機能をアドバタイズします。**disable** オプションは、追加パス受信機能のアドバタイズをディセーブルにします。 このコマンドの **no** 形式は、追加パスの受信機能をディセーブルにします。
Step 3	**show bgp neighbor** 例： `switch(config-router-neighbor-af)# show bgp neighbor`	ローカル ピアがリモート ピアへの追加パス送受信機能をアドバタイズしたかを表示します。

例

BGP ピアに追加のパスを送受信する機能をアドバタイズする BGP の設定例を示します。

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# neighbor 10.131.31.2 remote-as 100
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# capability additional-paths send
switch(config-router-neighbor-af)# capability additional-paths receive
```

追加パスの送受信の設定

BGP ピア間の追加パスの送受信機能を設定できます。これを行うには、アドレス ファミリ設定モードで次のコマンドを使用します。

手順の概要

1. [**no**] **additional-paths send**
2. [**no**] **additional-paths receive**
3. **show bgp neighbor**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	[**no**] **additional-paths send** 例： `switch(config-router-af)# additional-paths send`	機能が無効になっていないこのアドレスファミリで、すべてのネイバーの追加パスの送信機能を有効にします。 このコマンドの **no** 形式を使用すると、送信機能が無効になります。
Step 2	[**no**] **additional-paths receive** 例： `switch(config-router-af)# additional-paths receive`	機能が無効になっていないこのアドレスファミリで、すべてのネイバーの追加パスの受信機能を有効にします。 このコマンドの **no** 形式を使用すると、受信機能が無効になります。
Step 3	**show bgp neighbor** 例： `switch(config-router-af)# show bgp neighbor`	ローカル ピアがリモート ピアへの追加パス送受信機能をアドバタイズしたものとして表示します。

例

機能が無効になっていない指定されたアドレス ファミリで、すべてのネイバーの追加パスの受信機能を有効にする例を示します。

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# additional-paths send
switch(config-router-af)# additional-paths receive
```

アドバタイズされるパスの設定

BGP にアドバタイズされたパスを指定できます。これを行うには、ルート マップ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. [**no**] **set ip next-hop unchanged**
2. [**no**] **set path-selection { all | backup | best2 | multipaths}** | advertise
3. **show bgp** {**ipv4** | **ipv6**} **unicast** [*ip-address* | *ipv6-prefix*] [**vrf** *vrf-name*]

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	[**no**] **set ip next-hop unchanged** 例： `switch(config-route-map)# set ip next-hop unchanged`	不変のネクストホップ IP アドレスを指定します。
Step 2	[**no**] **set path-selection { all | backup | best2 | multipaths}** | advertise 例： `switch(config-route-map)# set path-selection all advertise`	すべてのパスが指定されたプレフィックスにアドバタイズされるように指定します。次のいずれかのオプションを選択できます。 • all：使用可能なすべての有効なパスをアドバタイズします。 • backup：バックアップ パスとしてマークされたパスをアドバタイズします。このオプションでは、additional-path install backup コマンドを使用してバックアップ パスを有効にする必要があります。 • best2：2 番目に最適なパスをアドバタイズします。これは、すでに計算されているベスト パスを除き、残りの使用可能なパスのベスト パスです。 • multipaths：すべてのマルチパスをアドバタイズします。このオプションでは、maximum-paths コマンドを使用してマルチパスを有効にする必要があります。 （注） マルチパスがない場合、backup オプションと best2 オプションは同じです。マルチパスがある場合、best2 はマルチパスのリストの最初のパスで、バックアップは計算されたベストパスとマルチパスを除くすべての使用可能なパスのベストパスです。 このコマンドの **no** 形式は、最適パスだけがアドバタイズされるように指定します。
Step 3	**show bgp** {**ipv4** | **ipv6**} **unicast** [*ip-address* | *ipv6-prefix*] [**vrf** *vrf-name*] 例：	プレフィックスの追加パスのパス ID とこれらのパスのアドバタイズメント情報を表示します。

	コマンドまたはアクション	目的
	`switch(config-route-map)# show bgp ipv4 unicast`	

例

すべてのパスがプレフィックス リスト p1 にアドバタイズされるよう指定する例を示します。

```
switch# configure terminal
switch(config)# route-map PATH_SELECTION_RMAP
switch(config-route-map)# match ip address prefix-list p1
switch(config-route-map)# set path-selection all advertise
```

追加パス選択の設定

プレフィックスに追加のパスを選択する機能を設定できます。これを行うには、アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. [**no**] **additional-paths selection route-map** *map-name*
2. {| } [ip-address | ipv6-prefix] [vrf-name] **show bgpipv4ipv6unicastvrf**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	[**no**] **additional-paths selection route-map** *map-name* 例： `switch(config-router-af)# additional paths selection route-map map1`	プレフィックスに追加のパスを選択する機能を設定します。 このコマンドの **no** 形式は、追加パス選択機能をディセーブルにします。
Step 2	{| } [ip-address	ipv6-prefix] [vrf-name] **show bgpipv4ipv6unicastvrf** 例： `switch(config-route-af)# show bgp ipv4 unicast`	プレフィックスの追加パスのパス ID とこれらのパスのアドバタイズメント情報を表示します。

例

指定されたアドレス ファミリで追加パス選択を設定する例を示します。

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# additional-paths selection route-map PATH_SELECTION_RMAP
```

eBGP の設定

eBGP シングルホップ チェックの無効化

シングルホップ eBGP ピアがローカル ルータに直接接続されているかどうかのチェック機能を無効にするように、eBGP を設定できます。このオプションは、直接接続されたスイッチ間のシングルホップ ループバック eBGP セッションの設定に使用します。

シングルホップ eBGP ピアが直接接続されているかどうかのチェックを無効にするには、ネイバー設定モードで次のコマンドを使用します。

手順の概要

1. **disable-connected-check**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**disable-connected-check** 例: `switch(config-router-neighbor)# disable-connected-check`	シングルホップ eBGP ピアが直接接続されているかどうかのチェックを無効にします。このコマンドの使用後、BGP セッションを手動でリセットする必要があります。

TTL セキュリティ ホップの構成

IP パケット ヘッダーの TTL 値が BGP ネイバー セッション用に設定された TTL 値以上の場合のみ BGP がセッションを確立または維持できるようにするには、次の作業を実行します。

始める前に

TTL セキュリティ チェックに対する BGP サポート機能の効果を最大化するために、参加している各ルータでこの機能を設定することを推奨します。この機能を有効にすると、eBGP セッションが受信方向だけ保護され、送信 IP パケットまたはリモート ルータは影響を受けません。

（注）

- TTL セキュリティ チェックに対する BGP サポート機能がマルチホップ ネイバー セッション用に構成されている場合、**neighbor ebgp-multihop** コマンドは必要なく、この機能を構成する前にこのコマンドを無効にする必要があります。
- 大きい直径のマルチホップ ピアリングでは、TTL セキュリティ チェックに対する BGP サポート機能の効果は下がります。大きい直径のピアリング用に設定された BGP ルータに対する CPU 利用率に基づく攻撃の場合は、影響を受けたネイバー セッションをシャットダウンして、この攻撃に対処する必要がある場合があります。
- この機能は、ローカルネットワークおよびリモートネットワーク内部が損なわれているピアからの攻撃には効果的ではありません。この制約事項には、ローカル ネットワークとリモート ネットワークの間のネットワーク セグメント上のピアも含まれます。

手順の概要

1. **enable**
2. **trace** *[protocol] destination*
3. **configure terminal**
4. **router bgp** *autonomous-system-number*
5. **neighbor** *ip-address*
6. **ttl-security hops** *hop-count*
7. **end**
8. **show running-config**
9. **show ip bgp neighbors** *[ip-address]*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**enable** 例： `switch(config)# enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたら、パスワードを入力します。
Step 2	**trace** *[protocol] destination* 例： `switch(config)# trace ip 10.1.1.1`	パケットが宛先に移動中、実際に通過する指定されたプロトコルのルートを検出します。 trace コマンドを入力して、指定されたピアへのホップ カウントを決定します。
Step 3	**configure terminal** 例： `switch(config)# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
Step 4	**router bgp** *autonomous-system-number* 例： `switch(config)# router bgp 65000`	ルータ コンフィギュレーション モードを開始して、BGP ルーティング プロセスを作成します。
Step 5	**neighbor** *ip-address* 例： `switch(config)# neighbor 10.1.1.1`	ネイバー IP アドレスを構成します。
Step 6	**ttl-security hops** *hop-count* 例： `switch(config)# ttl-security hops 2`	2 つのピアを区切るホップの最大数を設定します。 hop-count 引数は、ローカル ピアとリモート ピアを区切るホップ カウントに設定されます。IP パケット ヘッダーの予想される TTL 値が 254 の場合、数値 1 を hop-count 引数に設定する必要があります。値の範囲は、1 ～ 254 の数番です。 TTL セキュリティ チェックに対する BGP サポート機能が有効な場合、BGP は、予想値以上の TTL 値を持つ着信 IP パケットを受け入れます。受け入れられないパケットは廃棄されます。 この設定例では、予想される着信 TTL 値が 253（255 引く TTL 値の 2）以上に設定されます。これは、BGP ピアから予想される最小 TTL 値です。ローカル ルータは、10.1.1.1 ネイバーが 1 または 2 ホップ離れている場合だけ、このネイバーからのピアリングセッションを受け入れます。
Step 7	**end** 例： `switch(config)# end`	ルータ コンフィギュレーション モードを終了して、特権 EXEC モードを開始します。
Step 8	**show running-config** 例： `switch(config)# show running-config | begin bgp`	（任意）現在実行中のコンフィギュレーション ファイルの内容を表示します。 このコマンドの出力は、各ピアの neighbor ttl-security コマンドの構成を出力の BGP コンフィギュレーション セクションの下に表示します。そのセクションには、ネイバーアドレスおよび構成されたホップ カウントが含まれます。 （注） この例では、このタスクに適用可能な構文だけが使用されています。詳細については、『Cisco IOS IP Routing: BGP Command Reference』を参照してください。

	コマンドまたはアクション	目的
Step 9	**show ip bgp neighbors** *[ip-address]* 例： `switch(config)# show ip bgp neighbors 10.4.9.5`	（任意）ネイバーへの TCP 接続および BGP 接続の情報を表示します。 このコマンドは、TTL セキュリティ チェックに対する BGP サポート機能が有効になっている場合、「External BGP neighbor may be up to number hops away」と表示します。この number 値は、ホップ カウントを表します。これは、1 ～ 254 の数値です。 （注） この例では、このタスクに適用可能な構文だけが使用されています。詳細については、『Cisco IOS IP Routing: BGP Command Reference』を参照してください。

eBGP マルチホップの設定

eBGP マルチホップをサポートする eBGP 存続可能時間（TTL）値を設定できます。eBGP ピアは状況によって、別の eBGP ピアに直接接続されず、リモート eBGP ピアに到達するために複数のホップを必要とします。ネイバー セッションに eBGP TTL 値を設定すると、このようなマルチホップ セッションが可能になります。

（注） この設定は、BGP インターフェイス ピアリングではサポートされません。

eBGP マルチホップを設定するには、ネイバー コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **ebgp-multihop** *ttl-value*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**ebgp-multihop** *ttl-value* 例： `switch(config-router-neighbor)# ebgp-multihop 5`	eBGP マルチホップの eBGP TTL を設定します。有効な範囲は 2 ～ 255 です。このコマンドの使用後、BGP セッションを手動でリセットする必要があります。

高速外部フォールオーバーの無効化

Cisco NX-OS デバイスは、すべての VRF のネイバーおよびアドレス ファミリ（IPv4 または IPv6）の高速外部フォールオーバーをデフォルトでサポートします。通常、BGP ルータと直接接続 eBGP ピア間の接続が失われると、ピアとの eBGP セッションをリセットすることによって、BGP が高速外部フォールオーバーを開始します。この高速外部フォールオーバーをディセーブルにすると、リンク フラップが原因の不安定さを制限できます。

高速外部フォールオーバーをディセーブルにするには、ルータ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **no fast-external-fallover**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**no fast-external-fallover** 例： `switch(config-router)# no fast-external-fallover`	eBGP ピアの高速外部フォールオーバーをディセーブルにします。このコマンドは、デフォルトでイネーブルになっています。

AS パス属性の制限

AS パス属性で自律システム番号が高いルートを廃棄するように eBGP を設定できます。

AS パス属性で AS 番号の多いルートを廃棄するには、ルータ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **maxas-limit** *number*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**maxas-limit** *number* 例： `switch(config-router)# maxas-limit 50`	AS パス セグメントの番号が指定された上限を超えている eBGP ルートを廃棄します。指定できる範囲は 1 ～ 2000 です。

ローカル AS サポートの設定

ローカル AS 機能では、ルータが実際の AS に加えて、2 番めの自律システム（AS）のメンバであるように見せることができます。ローカル AS を使用すると、ピアリングの調整を変更せずに 2 つの ISP をマージできます。マージされた ISP 内のルータは、新しい自律システムのメンバになりますが、使用者に対しては古い自律システム番号を使用し続けます。

この機能は、正しい eBGP ピアにしか使用できません。別のコンフェデレーションのサブ自律システムのメンバである 2 ピアに対しては、この機能は使用できません。

さらに、remote-asコマンドで設定されたリモートピアのASNは、local-asコマンドで設定されたローカルデバイスのASNと同一にすることはできません。

eBGP ローカル AS のサポートを設定するには、ネイバー コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **local-as** *number* [**no-prepend** [**replace-as** [**dual-as**]]]

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**local-as** *number* [**no-prepend** [**replace-as** [**dual-as**]]] 例： `switch(config-router-neighbor)# local-as 1.1`	AS_PATH 属性にローカル AS の *number* を付加するよう eBGP を設定します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

例

次に、VRF のローカル AS サポートを設定する例を示します。

```
switch# configure terminal
switch(config)# router bgp 1
switch(config-router)# vrf test
switch(config-router-vrf)# local-as 1
switch(config-router-vrf)# show running-config bgp
```

AS 連合の設定

AS 連合を設定するには、連合識別情報を指定する必要があります。AS 連合内の自律システム グループは、自律システム番号として連合 ID を持つ、1 つの自律システムとして外部で認識されます。

BGP 連合 ID を設定するには、ルータ設定モードで次のコマンドを使用します。

手順の概要

1. **confederation identifier** *as-number*
2. **bgp confederation peers** *as-number* [*as-number2...*]

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**confederation identifier** *as-number* 例： `switch(config-router)# confederation identifier 4000`	ルータ設定モードで、このコマンドは BGP 連合 ID を設定します。 このコマンドによって、BGP ネイバーセッションの自動通知およびセッションリセットが開始されます。
Step 2	**bgp confederation peers** *as-number* [*as-number2...*] 例： `switch(config-router)# bgp confederation peers 5 33 44`	ルータ設定モードで、このコマンドは AS 連合に属する自律システムを設定します。 このコマンドは、連合に属する自律システムのリストを指定し、BGP ネイバーセッションの自動通知とセッションリセットをトリガーします。

ルート リフレクタの設定

ルート リフレクタとして動作するローカル BGP スピーカに対するルート リフレクタ クライアントとして、iBGP ピアを設定できます。ルート リフレクタとそのクライアントがともにクラスタを形成します。クライアントからなるクラスタには通常、ルート リフレクタが 1 つ存在します。このような状況では、ルート リフレクタのルータ ID でクラスタを識別します。ネットワークの冗長性を高め、シングルポイント障害を回避するために、複数のルートリフレクタからなるクラスタを設定できます。クラスタ内のすべてのルート リフレクタは、同じ 4 バイト クラスタ ID で設定する必要があります。これは、ルートリフレクタが同じクラスタ内のルートリフレクタからのアップデートを認識できるようにするためです。

始める前に

BGPをイネーブルにする必要があります。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **cluster-id** *cluster-id*
4. **address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}

5. （任意） **client-to-client reflection**
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
9. **route-reflector-client**
10. （任意） **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors**
11. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例： `switch# configure terminal`	グローバル コンフィギュレーション モードを開始します。

Step 2	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		BGP モードを開始し、ローカル BGP スピーカに自律システム番号を割り当てます。

Step 3	**cluster-id** *cluster-id* 例： `switch(config-router)# cluster-id 192.0.2.1`	クラスタに対応するルート リフレクタの 1 つとして、ローカルルータを設定します。クラスタを識別するクラスタ ID を指定します。このコマンドによって、BGP ネイバー セッションの自動ソフト クリアまたはリフレッシュが開始されます。

Step 4	**address-family** {**ipv4**	**ipv6**} {**unicast**	**multicast**}
			例：
			`switch(config-router)# address-family ipv4 unicast`
			`switch(config-router-af)#`
				指定のアドレス ファミリに対応するグローバル アドレス ファミリ コンフィギュレーション モードを開始します。

Step 5	（任意） **client-to-client reflection** 例： `switch(config-router-af)# client-to-client reflection`	クライアント間のルートリフレクションを設定します。この機能は、デフォルトでイネーブルになっています。このコマンドによって、BGP ネイバーセッションの自動ソフトクリアまたはリフレッシュが開始されます。

Step 6	**exit**
	例：
	`switch(config-router-af)# exit`
	`switch(config-router)#`
		ルータ アドレス コンフィギュレーション モードを終了します。

	コマンドまたはアクション	目的

Step 7	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.0.2.10 remote-as 65535`
	`switch(config-router-neighbor)#`
		リモート BGP ピアの IP アドレスおよび AS 番号を設定します。

Step 8	**address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		ユニキャスト IPv4 アドレス ファミリに対応するネイバー アドレス ファミリ コンフィギュレーション モードを開始します。

Step 9	**route-reflector-client** 例： `switch(config-router-neighbor-af)# route-reflector-client`	BGP ルート リフレクタとしてデバイスを設定し、そのクライアントとしてネイバーを設定します。このコマンドによって、BGP ネイバー セッションの自動通知およびセッション リセットが開始されます。
Step 10	（任意） **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors** 例： `switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors`	BGP ピアを表示します。
Step 11	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、ルート リフレクタとしてルータを設定し、クライアントとしてネイバーを 1 つ追加する例を示します。

```
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
```

アウトバウンド ルート マップを使用した、反映されたルートのネクスト ホップの設定

アウトバウンド ルート マップを使用して、BGP ルート リフレクタの反映されたルートのネクスト ホップを変更できます。ネクスト ホップ アドレスとしてピアのローカル アドレスを指定するため、アウトバウンド ルート マップを設定できます。

（注） この項で説明している **next-hop-self** コマンドは、ルート リフレクタによってクライアントに反映されるルートに対してこの機能を有効にしません。この機能は、アウトバウンドルートマップを使用した場合にだけ有効にできます。

始める前に

BGP を有効にする必要があります（「BGP の有効化」の項を参照）。

正しい VDC を使用していることを確認します（または **switchto vdc** コマンドを使用します）。

set next-hop を入力する必要があります コマンドを入力して、アドレス ファミリ固有のネクスト ホップ アドレスを設定する必要があります。たとえば、IPv6アドレスファミリの場合は、 **set ipv6 next-hop peer-address** コマンドを入力する必要があります。

- ルートマップを使用して IPv4 ネクストホップを設定する場合： **set ip next-hop peer-address** がルートマップと一致する場合、ネクストホップはピアのローカル アドレスに設定されます。ネクスト ホップがルート マップで設定されていない場合、ネクスト ホップはパスに保存されているネクスト ホップに設定されます。
- ルートマップを使用してIPv6ネクストホップを設定する場合： **set ipv6 next-hop peer-address** がルートマップと一致する場合、ネクスト ホップは次のように設定されます。
  - IPv6 ピアでは、ネクスト ホップはピアのローカル IPv6 アドレスに設定されます。
  - IPv4 ピアの場合、 **update-source** が設定されている場合、ネクスト ホップは、該当する場合、発信元インターフェイスの IPv6 アドレスに設定されます。IPv6 アドレスが設定されていない場合、ネクスト ホップは設定されません。
  - IPv4 ピアの場合、 **update-source** が設定されていない場合、ネクスト ホップは、該当する場合、送信先インターフェイスの IPv6 アドレスに設定されます。IPv6 アドレスが設定されていない場合、ネクスト ホップは設定されません。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ip-address* **remote-as** *as-number*

4. （任意） **update-source** *interface number*
5. **address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
6. **route-reflector-client**
7. **route-map** *map-name* **out**
8. （任意） **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} [**ip-address** | **ipv6-prefix**] **route-map** *map-name* [**vrf** *vrf-name*]
9. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 200`
	`switch(config-router)#`
		BGP モードを開始し、ローカル BGP スピーカに自律システム番号を割り当てます。

Step 3	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.0.2.12 remote-as 200`
	`switch(config-router-neighbor)#`
		リモート BGP ピアの IP アドレスおよび AS 番号を設定します。

Step 4	（任意） **update-source** *interface number* 例： `switch(config-router-neighbor)# update-source loopback 300`	BGP セッションの送信元を指定し、更新します。

Step 5	**address-family** {**ipv4**	**ipv6**} {**unicast**	**multicast**}
			例：
			`switch(config-router-neighbor)# address-family ipv4 unicast`
			`switch(config-router-neighbor-af)#`
				指定のアドレス ファミリに対応するグローバル アドレス ファミリ コンフィギュレーション モードを開始します。

Step 6	**route-reflector-client** 例： `switch(config-router-neighbor-af)# route-reflector-client`	BGP ルート リフレクタとしてデバイスを設定し、そのクライアントとしてネイバーを設定します。このコマンドによって、BGP ネイバー セッションの自動通知およびセッション リセットが開始されます。

	コマンドまたはアクション	目的
Step 7	**route-map** *map-name* **out** 例： `switch(config-router-neighbor-af)# route-map setrrnh out`	発信ルートに設定された BGP ポリシーを適用します。
Step 8	（任意） **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} [**ip-address** | **ipv6-prefix**] **route-map** *map-name* [**vrf** *vrf-name*] 例： `switch(config-router-neighbor-af)# show bgp ipv4 unicast route-map setrrnh`	ルートマップと一致する BGP ルートを表示します。
Step 9	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor-af)# copy running-config startup-config`	この設定変更を保存します。

例

アウトバウンドルートマップを使用して、BGP ルート リフレクタの反映されたルートのネクスト ホップを設定する例を示します。

```
switch(config)# interface loopback 300
switch(config-if)# ip address 192.0.2.11/32
switch(config-if)# ipv6 address 2001::a0c:1a65/64
switch(config-if)# ip router ospf 1 area 0.0.0.0
switch(config-if)# exit
switch(config)# route-map setrrnh permit 10
switch(config-route-map)# set ip next-hop peer-address
switch(config-route-map)# exit
switch(config)# route-map setrrnhv6 permit 10
switch(config-route-map)# set ipv6 next-hop peer-address
switch(config-route-map)# exit
switch(config)# router bgp 200
switch(config-router)# neighbor 192.0.2.12 remote-as 200
switch(config-router-neighbor)# update-source loopback 300
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)# route-map setrrnh out
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# address-family ipv6 unicast
switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)# route-map setrrnhv6 out
```

ルート ダンプニングの設定

iBGP ネットワーク上でのルート フラップの伝播を最小限に抑えるために、ルート ダンプニングを設定できます。

ルート ダンプニングを設定するには、アドレス ファミリまたは VRF アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **dampening** [{*half-life reuse-limit suppress-limit max-suppress-time* | **route-map** *map-name*}]

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**dampening** [{*half-life reuse-limit suppress-limit max-suppress-time* | **route-map** *map-name*}] 例： `switch(config-router-af)# dampening route-map bgpDamp`	機能ネゴシエーションをディセーブルにします。パラメータ値は次のとおりです。 • *half-life*：指定できる範囲は 1 ～ 45 です。 • *resuse-limit*指定できる範囲は 1 ～ 20000 です。 • *suppress-limit*：指定できる範囲は 1 ～ 20000 です。 • *max-suppress-time*：指定できる範囲は 1 ～ 20000 です。

ロード シェアリングおよび ECMP の設定

等コスト マルチパス ロード バランシング用に BGP がルート テーブルに追加するパスの最大数を設定できます（EXMP）。

パスの最大数を設定するには、ルータ アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **maximum-paths** [**ibgp**] *maxpaths*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**maximum-paths** [**ibgp**] *maxpaths* 例： `switch(config-router-af)# maximum-paths 8`	ロードシェアリング用の等コストパスの最大数を設定します。デフォルトは 1 です。

BGP 経由不等コストマルチパス（UCMP）

UCMP は加重 ECMP とも呼ばれます。これは、ネクストホップごとに異なる重みを持つ、同じ宛先への複数のルートを許可し、ルーティングされたトラフィックをそれらの複数のネクストホップにロードバランシングするメカニズムです。基本的なUCMPは、ほとんどの顧客の要件に対応します。負荷エントロピーは、リンク使用効率を最大化する最良の方法です。

多くの場合、ネットワーク内のアプリケーションの分散は不均衡になりがちです。新しいクラスタは、古いクラスタとは異なるオーバーサブスクリプション率でロールインします。新しいクラスタには、古いクラスタよりも強力なサーバーがあり、CPU ごとにより多くの負荷を処理できます。ネットワークは完全ではないため、ルーティング動作をある程度制御する必要があります。トラフィックの負荷を分散し、ルーティング動作の制御を管理するために、BGP 経由の加重 ECMP を構成できます。

（注） リンク帯域幅拡張コミュニティは、非推移的な属性として定義されていますが、eBGP セッション全体でアドバタイズする必要があります。

Next-hop-self は、アドバタイズから Link-Bandwidth Extended Community を取り除く必要があります。

UCMP over BGP の有効化

ユースケースでリソースの不均等な分散と最適ではないトラフィック分散が発生している場合の解決策は、BGP 上で重み付き ECMP を構成することです。各インスタンスの重みは、（ホストまたはコントローラーから）ルートを挿入して通知できます。その後、インフラストラクチャ全体の重みを集計し、アプリケーション展開の分布に比例するようにトラフィックを配信できます。

BGP 経由 UCMP の注意事項と制限事項

- BGPは、draft-ietf-idr-link-bandwidth-06.txtで定義されているリンク帯域幅拡張コミュニティを使用して、重み付けECMP機能を実装します。リンク帯域幅拡張コミュニティは、次ホップが変更されていない限り、非推移的な属性として定義されていますが、eBGPセッション全体でアドバタイズされます。
- iBGP ピアと eBGP ピアの両方からリンク帯域幅拡張コミュニティを受け入れることができます。
- 重み付けプログラミングの場合、リンク帯域幅拡張コミュニティには、RIB にダウンロードする前に 0 ～ 1000 の間で正規化された 4 バイトの浮動小数点整数としてバイト/秒でエンコードされたリンク帯域幅があります。
- ハードウェア ECMP 幅は 64 サイズに固定されています。

最大プレフィックス数の設定

BGPがBGPピアから受け取ることのできるプレフィックスの最大数を設定できます。任意で、プレフィックス数がこの値を超えた場合に、BGP に警告メッセージを生成させる、またはピアとの BGP セッションを切断させることを設定できます。

BPG ピアに認めるプレフィックスの最大数を設定するには、ネイバーアドレスファミリ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **maximum-prefix** *maximum* [*threshold*] [**restart** *time* | **warning-only**]

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**maximum-prefix** *maximum* [*threshold*] [**restart** *time* | **warning-only**] 例： `switch(config-router-neighbor-af)# maximum-prefix 12`	ピアからのプレフィックスの最大数を設定します。パラメータの範囲は次のとおりです。 • *maximum*：指定できる範囲は 1 ～ 300000 です。 • *threshold*：指定できる範囲は 1 ～ 100 % です。デフォルトは 75% です。 • *time*：指定できる範囲は 1 ～ 65535 分です。

	コマンドまたはアクション	目的
		このコマンドによって、プレフィックス限度を超えた場合に、BGP ネイバーセッションの自動通知およびセッション リセットが開始されます。

DSCP の設定

ネイバーの differentiated services code point（DSCP）を設定します。IPv4 または IPv6 のローカル発信パケットの DSCP 値を指定できます。

DSCP 値を設定するには、ネイバー コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **dscp** *dscp_value*

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**dscp** *dscp_value*
	例：
	`switch(config-router-neighbor)# dscp 63`
	次に、対応する **show** コマンドの例を示します。
	`show ipv6 bgp neighbors`
	`BGP neighbor is 10.1.1.1, remote AS 0, unknown link, Peer index 4`
	`  BGP version 4, remote router ID 0.0.0.0`
	`  BGP state = Idle, down for 00:13:34, retry in 0.000000`
	`  DSCP (DiffServ CodePoint): 0`
	`  Last read never, hold time = 180, keepalive interval is 60 seconds`
		ネイバーの Differentiated Services Code Point（DSCP）の値を設定します。DSCP 値には、0 ～ 63 の数字、または、**ef**、**af11**、**af12**、**af13**、**af21**、**af22**、**af23**、**af31**、**af32**、**af33**、**af41**、**af42**、**af43**、**cs1**、**cs2**、**cs3**、**cs4**、**cs5**、**cs6**、または **cs7** のいずれかのキーワードを指定できます。
		デフォルト値は cs6 です。

ダイナミック機能の設定

BGP ピアのダイナミック機能を設定できます。

ダイナミック機能を設定するには、ネイバーコンフィギュレーションモードで次のコマンドを使用します。

手順の概要

1. **dynamic-capability**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**dynamic-capability** 例： `switch(config-router-neighbor)# dynamic-capability`	ダイナミック機能をイネーブルにします。このコマンドによって、BGP ネイバー セッションの自動通知およびセッション リセットが開始されます。

集約アドレスの設定

BGP ルート テーブルの集約アドレス エントリを設定できます。

集約アドレスを設定するには、ルータ アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **aggregate-address** *ip-prefix/length* [**as-set**] [**summary-only**] [**advertise-map** *map-name*] [**attribute-map** *map-name*] [**suppress-map** *map-name*]

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**aggregate-address** *ip-prefix/length* [**as-set**] [**summary-only**] [**advertise-map** *map-name*] [**attribute-map** *map-name*] [**suppress-map** *map-name*] 例： `switch(config-router-af)# aggregate-address 192.0.2.0/8 as-set`	集約アドレスを作成します。このルートに関してアドバタイズされるパスは、集約されているすべてのパスに含まれるすべての要素からなる、自律システム セットです。 • **as-set** キーワードは、関係するパスから自律システムセットパス情報およびコミュニティ情報を生成します。 • **summary-only** キーワードは、アップデートから具体的なルートをすべてフィルタリングします。

	コマンドまたはアクション	目的
		• **advertise-map** キーワードおよび引数では、選択されたルートから属性情報を選択するためのルート マップを指定します。 • **attribute-map** キーワードおよび引数では、集約から属性情報を選択するためのルート マップを指定します。 • **suppress-map** キーワードおよび引数によって、固有性の強いルートを条件付きでフィルタリングします。BGP ルート集約の実行中に **suppress-map** オプションを指定すると、BGP ルート更新のコミュニティ属性を設定できます。このオプションを使用すると、より具体的なルートにコミュニティ属性を設定できます。 • **suppress-map** キーワードおよび引数によって、固有性の強いルートを条件付きでフィルタリングします。BGPルート集約の実行中に **suppress-map** オプションを指定すると、特定のより具体的なルートがピアにアドバタイズされないように抑制したり、suppress-map route-map 設定に応じて、いくつかのコミュニティ属性が設定されたより具体的なルートをアドバタイズしたりすることができます。match 句だけで設定されたルートマップは、一致基準を満たすより具体的なルートを抑制します。ただし、ルートマップが match および set 句で設定されている場合、一致基準を満たすルートは、ルートマップによって変更された適切な属性でアドバタイズされます。2 番目のオプションでは、より具体的なルートにコミュニティ属性を設定できます。

BGP ルートの抑制

新しく学習された BGP ルートが転送情報ベース（FIB）により確認され、ハードウェアでプログラミングされた後にのみ、これらのルートをアドバタイズするように Cisco NX-OS を設定できます。ルートがプログラミングされた後は、これらのルートに対する以降の変更にはこのハードウェア プログラミングのチェックは必要ありません。

BGP ルートを抑制するには、ルータ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **suppress-fib-pending**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**suppress-fib-pending** 例： `switch(config-router)# suppress-fib-pending`	新しく学習された BGP ルート（IPv4 または IPv6）がハードウェアでプログラミングされるまで、ダウンストリームの BGP ネイバーにアドバタイズされることを抑制します。

BGP 条件付きアドバタイズメントの設定

BGP がプロパゲートするルートを制限するように BGP 条件付きアドバタイズメントを設定できます。次の 2 つのルート マップを定義します。

- アドバタイズマップ：BGP が条件付きアドバタイズメントを考慮する前にルートが一致する必要のある条件を指定します。このルート マップには、適切な match 文を含めることができます。
- 存在マップまたは非存在マップ：BGP がアドバタイズ マップに一致するルートをプロパゲートする前に BGP テーブルに存在する必要のあるプレフィックスを定義します。非存在マップは、BGP がアドバタイズ マップに一致するルートをプロパゲートする前に BGP テーブルに存在してはならないプレフィックスを定義します。BGP は、これらのルート マップでプレフィックス リストの match 文内にある permit 文のみを処理します。
- Nexus は、条件付きルート アドバタイズメントを使用した他の BGP 属性変更操作（AS パスを付加する例）をサポートしていません。存在/非存在マップの構成に基づいてアドバタイズされるルートを制御するために使用されます。

ルートが条件を渡さない場合、そのルートが BGP テーブルにあれば BGP によってルートが取り消されます。

始める前に

BGP を有効にする必要があります（「BGP の有効化」のセクションを参照）。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ip-address* **remote-as** *as-number*
4. **address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}

5. **advertise-map** *adv-map* {**exist-map** *exist-rmap*|**non-exist-map** *nonexist-rmap*}
6. （任意） **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors**
7. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

Step 2	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		BGPモードを開始し、ローカルBGPスピーカに自律システム番号を割り当てます。

Step 3	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65534`
	`switch(config-router-neighbor)#`
		BGPルーティング用のネイバー設定モードを開始し、ネイバー IP アドレスを設定します。

Step 4	**address-family** {**ipv4**	**ipv6**} {**unicast**	**multicast**}
			例：
			`switch(config-router-neighbor)# address-family ipv4 multicast`
			`switch(config-router-neighbor-af)#`
				アドレス ファミリ設定モードを開始します。

Step 5	**advertise-map** *adv-map* {**exist-map** *exist-rmap*|**non-exist-map** *nonexist-rmap*} 例： `switch(config-router-neighbor-af)# advertise-map advertise exist-map exist`	2 つの設定済みルート マップに従い、ルートを条件付きでアドバタイズするようにBGPを設定します。 • *adv-map*：BGP がルートを次のルート マップに渡す前に、そのルートが渡す必要のある **match** 文を含むルート マップを指定します。*adv-map* には最大63文字の英数字を使用できます。大文字と小文字は区別されます。 • *exist-rmap*：プレフィックス リストの match ステートメントを使用してルート マップを指定します。BGPテーブル内のプレフィックスは、BGPがルートをアドバタイズする前に、プレフィックス リスト内のプレフィックスと一致する必要があります。*exist-rmap* には最大 63 文字の英数

	コマンドまたはアクション	目的
		字を使用できます。大文字と小文字は区別されます。 • *nonexist-rmap*：プレフィックス リストの match ステートメントを使用してルート マップを指定します。BGP テーブル内のプレフィックスは、BGP がルートをアドバタイズする前に、プレフィックス リスト内のプレフィックスと一致してはいけません。*nonexist-rmap* には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。 （注） BGP 条件付きアドバタイズメント機能の場合、exist マップまたは nonexist マップに関連付けられている場合、プレフィックス リストで「le」または「ge」ステートメントが使用されていないことを確認します。
Step 6	（任意） **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors** 例： `switch(config-router-neighbor-af)# show ip bgp neighbor`	BGP に関する情報、および設定した条件付きアドバタイズメントのルート マップに関する情報を表示します。
Step 7	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、BGP 条件付きアドバタイズメントを設定する例を示します。

```
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
```

ルートの再配布の設定

別のルーティングプロトコルからのルーティング情報を受け入れて、BGPネットワークを通じてその情報を再配布するように、BGP を設定できます。任意で、再配布ルートのためのデフォルトルートを割り当てることができます。

始める前に

BGPを有効にする必要があります。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **address-family** {**ipv4** | **ipv6** } {**unicast** | **multicast**}
4. **address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
5. **redistribute** {**direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static** | **icmpv6**} **route-map** *map-name*
6. （任意） **default-metric** *value*
7. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		BGPモードを開始し、ローカルBGPスピーカに自律システム番号を割り当てます。

Step 3	**address-family** {**ipv4**	**ipv6** } {**unicast**	**multicast**}
			例：
			`switch(config-router)# address-family vpnv4 unicast`
			`switch(config-router-af)#`
				アドレスファミリコンフィギュレーションモードを開始します。

	コマンドまたはアクション	目的

Step 4	**address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		アドレスファミリ コンフィギュレーション モードに入ります。

Step 5	**redistribute** {**direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static** | **icmpv6**} **route-map** *map-name* 例： `switch(config-router-af)# redistribute eigrp 201 route-map Eigrpmap`	他のプロトコルからのルートを BGP に再配布します。 Cisco NX-OS リリース 10.3(3)F 以降では、icmpv6 ルートを他のプロトコルから BGP に再配布するために **icmpv6** キーワードがサポートされています。
Step 6	（任意） **default-metric** *value* 例： `switch(config-router-af)# default-metric 33`	BGP へのデフォルト ルートを生成します。
Step 7	（任意） **copy running-config startup-config** 例： `switch(config-router-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、EIGRP を BGP に再配布する例を示します。

```
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
```

DMZ リンク帯域幅

DMZ リンク帯域幅機能は、複数の自律システム出口リンクを介して到達可能な BGP 学習ルートへのトラフィック ロード バランシングを有効にするために使用されます。ロード バランシングは、これらのリンクの帯域幅に比例して行われます。

リンク帯域幅拡張コミュニティは、直接接続された2つの（シングルホップ）eBGP ピア間のリンクの帯域幅を伝送するために使用されます。dmz-link-bandwidth コマンドがネイバーのアドレスファミリ モードで構成されている場合、Nexus デバイスは、直接接続された eBGP ネイバーから受信した BGP ルートにこの拡張コミュニティを接続します。この拡張コミュニティ属性は、send-community extended または send-community both コマンドで拡張コミュニティ交換が有効になっ

ているときに、iBGP ピアに伝達されます。この属性は、転送時に他のパスに関連するロードシェアリング値として使用されます。

さらに、BGP ピアから受信したルートのリンク帯域幅拡張通信を強制的に変更することもできます。また、ピアから受信したルートのサブセットのみにこの拡張コミュニティを構成することもできます。これを実現するには、ピアへのインバウンド ルートマップを構成し、その下に「set extcommunity bandwidth <1-4000000>」を構成します。

注意事項と制約事項

BGP DMZ リンク帯域幅

リンク帯域幅機能を構成する前に、次の注意事項と制約事項を考慮してください。

- **dmz-link-bandwidth** コマンドは、BGP ネイバーの IPv4 ユニキャストおよび IPv6 ユニキャスト アドレス ファミリでのみ設定できます。
- リンク帯域幅拡張コミュニティは、直接接続された BGP ネイバーから受信したルートにのみ接続されます。BGP マルチホップ ネイバーでは行われません。
- グローバル モードと VRF モードの両方で構成できます。
- この機能を有効にするには、**maximum-paths** コマンドを使用して、アドレスファミリで BGP マルチパスロードバランシングを構成する必要があります。
- リンク帯域幅拡張コミュニティのアドバタイズ先の iBGP ネイバー間で、BGP 拡張コミュニティ交換がイネーブルになっている必要があります。
- リンク帯域幅拡張コミュニティは、ある VRF から別の VRF にリークされたルートにシームレスに伝送されます。

BGP DMZ リンク帯域幅の構成

Cisco NX-OS リリース 10.5(1)F 以降では、この機能を構成できます。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **address-family [ipv4|ipv6] unicast**
4. **maximum-paths** *max-path*
5. **template peer** *peer-template-name*
6. **address-family [ipv4 | ipv6] unicast**
7. **dmz-link-bandwidth**
8. **neighbor** *neighbor*
9. **remote-as** *remote-as*
10. **address-family [ipv4 | ipv6] unicast**
11. **dmz-link-bandwidth**

12. **route-map** *name* **permit** *route*
13. **set extcommunity bandwidth <1-4000000>**
14. **neighbor** *neighbor* **address-family [ipv4 | ipv6] unicast route-map** *name* **in**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例： `switch# configure terminal`	グローバル コンフィギュレーション モードを開始します。
Step 2	**router bgp** *as-number* 例： `switch# configure terminal`	ルータ コンフィギュレーションモードを開始して、BGP ルーティング プロセスを作成または設定します。
Step 3	**address-family [ipv4|ipv6] unicast** 例： `switch(config)# address-family ipv4 unicast`	アドレス ファミリ IPv4 または IPv6 ユニキャストを構成します。
Step 4	**maximum-paths** *max-path* 例： `switch(config)# maximum-paths 10`	アドレスファミリで BGP マルチパスを有効にします。
Step 5	**template peer** *peer-template-name* 例： `switch(config)# template peer host_peer`	テンプレートモードを開始し、peer パラメータを構成します。
Step 6	**address-family [ipv4 | ipv6] unicast** 例： `switch(config)# address-family ipv4 unicast`	IPv4 または IPv6 のアドレス ファミリを構成します。
Step 7	**dmz-link-bandwidth** 例： `switch(config)# dmz-link-bandwidth`	リンク帯域幅拡張コミュニティを受信したルートに接続することで、直接接続されたピアへのトラフィックのロードバランシングを考慮するように BGP を構成します。

Step 8	**neighbor** *neighbor*
	例：
	`switch(config)# neighbor 1.1.1.1`
	`or`
	`switch(config)# neighbor 11::1`
		BGP ネイバーを構成します。

	コマンドまたはアクション	目的
Step 9	**remote-as** *remote-as* 例： `switch(config)# remote-as 100`	ネイバーの自律システム番号を指定します。
Step 10	**address-family [ipv4 | ipv6] unicast** 例： `switch(config)# address-family ipv4 unicast`	アドレス ファミリ IPv4 または IPv6 ユニキャストを構成します。
Step 11	**dmz-link-bandwidth** 例： `switch(config)# dmz-link-bandwidth`	リンク帯域幅拡張コミュニティを受信したルートに接続することで、直接接続されたピアへのトラフィックのロードバランシングを考慮するように BGP を設定します。
Step 12	**route-map** *name* **permit** *route* 例： `switch(config)# route-map change_link_bandwidth permit 10`	route-map を構成します。
Step 13	**set extcommunity bandwidth <1-4000000>** 例： `switch(config-route-map)# set extcommunity bandwidth 1000`	route-map を構成して、リンク帯域幅拡張コミュニティを設定しま

Step 14	**neighbor** *neighbor* **address-family [ipv4 | ipv6] unicast route-map** *name* **in**
	例：
	`switch(config-router)# neighbor 1.1.1.1`
	`switch (config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)# route-map change_link_bandwidth in`
		ネイバーを着信ルートマップに接続するように設定します。

BGP DMZ リンク帯域幅の構成例

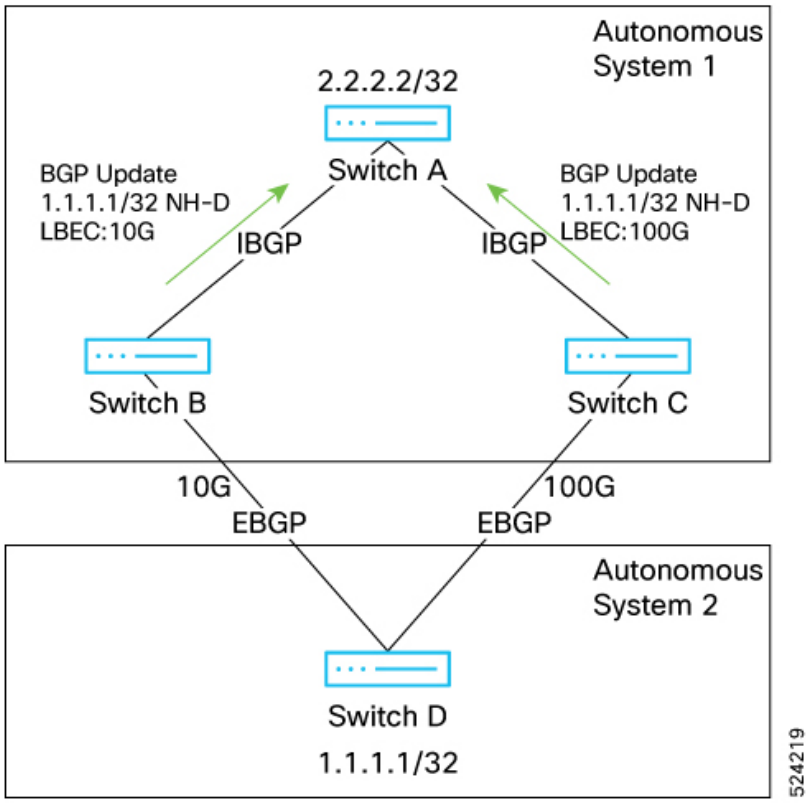
次の例では、AS 1 が 2 つの異なる帯域幅のリンクを介して AS 2 に接続されています。B-D リンクは 10G、C-D リンクは 100G です。B-D リンクは 10G、C-D リンクは 100G です。A-B とA-Cは IBGP セッションによって接続されています。EBGP セッションによって接続された B-D と C-D。

トラフィックのロードバランシングが、これらのリンク帯域幅に合わせて比例して行われるようにする場合（つまり、A から C に対し、B に対するより 10 倍の発信トラフィックを送信する場合）は、B と C で、EBGP ネイバー D に向けて dmz-link-bandwidth コマンドを実行します。B は B-D の帯域幅をリンク帯域幅拡張コミュニティ（Link Bandwidth Extended Community、LBEC）にパッケージ化し、ルートエントリ 1.1.1.1/32 の BGP パスにアタッチします。同様に、C は C-D の帯域幅を LBEC にパッケージ化し、ルートエントリ 1.1.1.1/32 の BGP パスにアタッチします。

B と C は、iBGP ピア A に LBEC とともにルートをアドバタイズします。

A では、BGP がハッシュ 10/110 を NH-B に、100/110 を NH-C に指定して、転送をプログラムします。

図 *33 : DMZ* リンク帯域幅の構成



スイッチ B から D および B から A の構成

```
router bgp 1
 neighbor D
    address-family ipv4|v6 unicast
      dmz-link-bandwidth
  neighbor A
    address-family ipv4|v6 unicast
       send-community extended
```

スイッチ C から D および C から A の構成

```
router bgp 1
 neighbor D
    address-family ipv4|v6 unicast
      dmz-link-bandwidth
  neighbor A
    address-family ipv4|v6 unicast
       send-community extended
```

スイッチ A のコントロールプレーンとデータ プレーンの状態

BGP テーブルの状態：

```
1.1.1.1/32
  NH-B    LBEC: 10G
  NH-C    LBEC: 100G
```

転送の状態：

```
1.1.1.1/32
  NH-B   hash 10:110
  NH-C   hash 100:110
```

リンク帯域幅拡張コミュニティを使用した不等コスト マルチパス（UCMP）の構成

始める前に

「BGP DMZ リンク帯域幅の構成」を参照してください。この機能を動作させるには、まずエッジデバイスでその機能を構成する必要があります。つまり、エッジデバイスでは、**dmz-link-bandwidth** コマンドを構成するか、 **set extcommunity link-bandwidth <1-4000000>** を使用してインバウンドルートマップを構成することによって、直接接続された ebgp ピアから受信した BGP ルートにリンク帯域幅拡張コミュニティを接続する必要があります。 それが発生するまで、このセクションで説明されている機能はいずれも動作しません。

Cisco NX-OS リリース 10.5(1)F 以降では、ネイバーのアドレスファミリ モードで、「link-bandwidth cumulative」コマンドが構成されている場合に、BGPスピーカーは直接接続されている BGP ネイバーにBGP 学習ルートまで利用可能な累積帯域幅を伝えることができます。これは、リンク帯域幅拡張コミュニティを活用することで実現されます。この拡張コミュニティに挿入された値 n で BGP ルートをアドバタイズします。ここで、n = min（使用可能なすべてのマルチパスのリンク帯域幅拡張コミュニティから取得した帯域幅の合計、アドバタイズメントが送信されるネイバーへのリンクの帯域幅）。

ガイドラインと制約事項

BGP UCMP 機能を構成する前に、次の注意事項と制約事項を考慮してください。

- **link-bandwidth cumulative** コマンドは、BGP ネイバーの IPv4 ユニキャストおよび IPv6 ユニキャスト アドレス ファミリでのみ構成できます。
- 直接接続された BGP ネイバーに対してのみ有効になります。
- このコマンドは、使用可能なすべてのマルチパスにリンク帯域幅拡張コミュニティがある場合にのみ有効です。
- グローバル モードと VRF モードの両方で構成できます。
- リンク帯域幅拡張コミュニティは、ある VRF から別の VRF にリークされたルートにシームレスに伝送されます。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *neighbor*
4. **remote-as** *remote-as*
5. **address-family [ipv4 | ipv6] unicast**
6. **send-community extended**
7. **link-bandwidth cumulative**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例： `switch# configure terminal`	グローバル コンフィギュレーション モードを開始します。
Step 2	**router bgp** *as-number* 例： `switch(config)# router bgp 120`	ルータ コンフィギュレーション モードを開始して、BGP ルーティング プロセスを作成または設定します。

Step 3	**neighbor** *neighbor*
	例：
	`switch(config)# neighbor 1.1.1.1`
	`or`
	`switch(config)# neighbor 11::1`
		BGP ネイバーを構成します。

Step 4	**remote-as** *remote-as* 例： `switch(config)# remote-as 100`	ネイバーの自律システム番号を指定します。
Step 5	**address-family [ipv4 | ipv6] unicast** 例： `switch(config)# address-family ipv4 unicast`	アドレス ファミリ IPv4 または IPv6 ユニキャストを構成します。
Step 6	**send-community extended** 例： `switch(config)# send-community extended`	このネイバーへの BGP 拡張コミュニティの送信を構成します。
Step 7	**link-bandwidth cumulative** 例：	累積リンク帯域幅をネイバーに送信します。この構成では、マルチパスのいずれかにリンク帯域幅拡張

	コマンドまたはアクション	目的
	`switch(config)# link-bandwidth cumulative`	コミュニティがない場合、累積リンク帯域幅はネイバーにアドバタイズされません。

設定例

設定例

次の図では、以下の点が描かれています：

- 4 つのクロスレイヤで直接接続された ebgp デバイス間のシングルホップ。
- すべてのリンクは 100G ですが、10G である B-D と B-E を除きます。これは、B から接続先 1.1.1.1 へのトラフィックが 20G のリンク帯域幅のためボトルネックになることを意味します。
- ユーザーがエンドツーエンドのトラフィックのロードバランスを望む場合、このリンク帯域幅の狭さを考慮する必要があります。
- レイヤ T1 の各デバイスで、レイヤ T0 のすべての BGPネイバーに対してコマンド **dmz-link-bandwidth** を実行します。
- レイヤ T1 の各デバイスで、レイヤ T2 のすべての BGP ピアに対してコマンド **link-bandwidth cumulative** を実行します。
- レイヤ T2 の各デバイスで、レイヤ T1 および T3 のすべての BGP ピアに対してコマンド **link-bandwidth cumulative** を実行します。
- レイヤ T3 の各デバイスで、レイヤ T2 のすべての BGP ピアに対してコマンド **link-bandwidth cumulative** を実行します。
- コマンド **dmz-link-bandwidth** により、スイッチ D はリンク D-F の帯域幅をリンク帯域幅拡張コミュニティ（LBEC）にパッケージ化し、ルート エントリ 1.1.1.1/32 の対応する BGP 経路にアタッチします。
- コマンド **dmz-link-bandwidth** により、スイッチ E はリンク E-F の帯域幅を LBEC にパッケージ化し、ルートエントリ 1.1.1.1/32 の対応する BGP 経路にアタッチします。
- コマンド **link-bandwidth cumulative** により、スイッチ A、B、C、D、および E は帯域幅 **n** を LBEC に挿入し、一方、BGP の更新をそれが構成されているピアにアドバタイズします。
- 値 **n** は、最小値を求める式（すべてのマルチパスの LBEC の合計、アップデートがアドバタイズされるピアへの帯域幅）を使用して計算されます。
- BGP アップデートでの LBEC の伝播により、すべてのデバイスでの転送が比例ハッシュを用いてプログラムされます。
- リンク障害が発生した場合、LBEC は動的に再計算されます。

図 *34* : 構成例 *1*

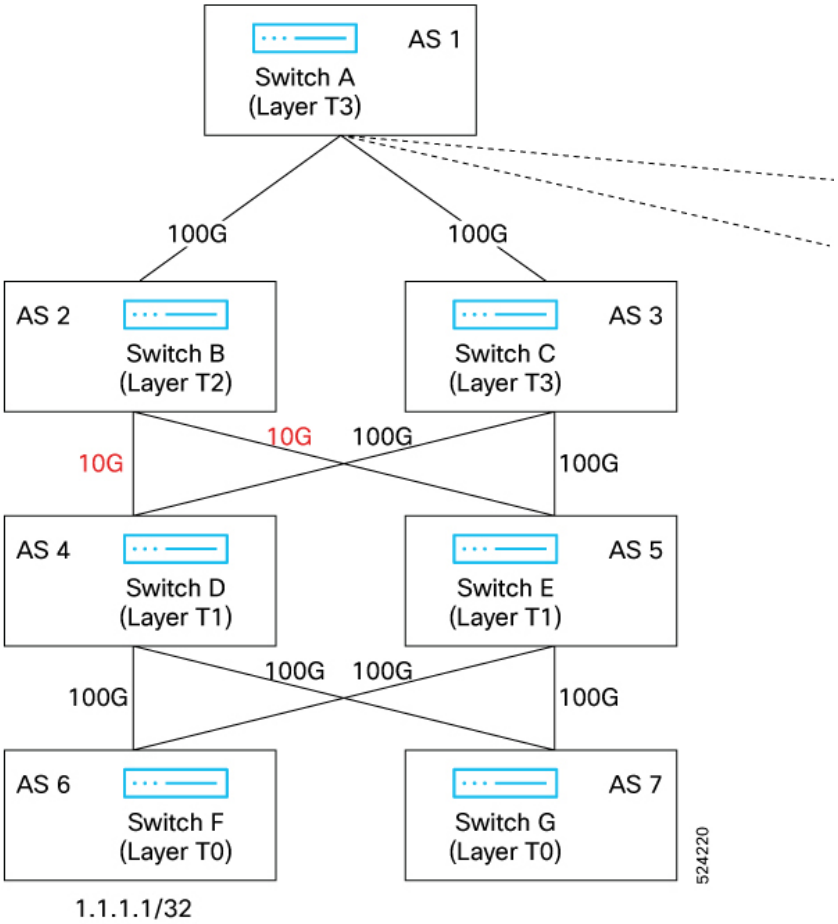
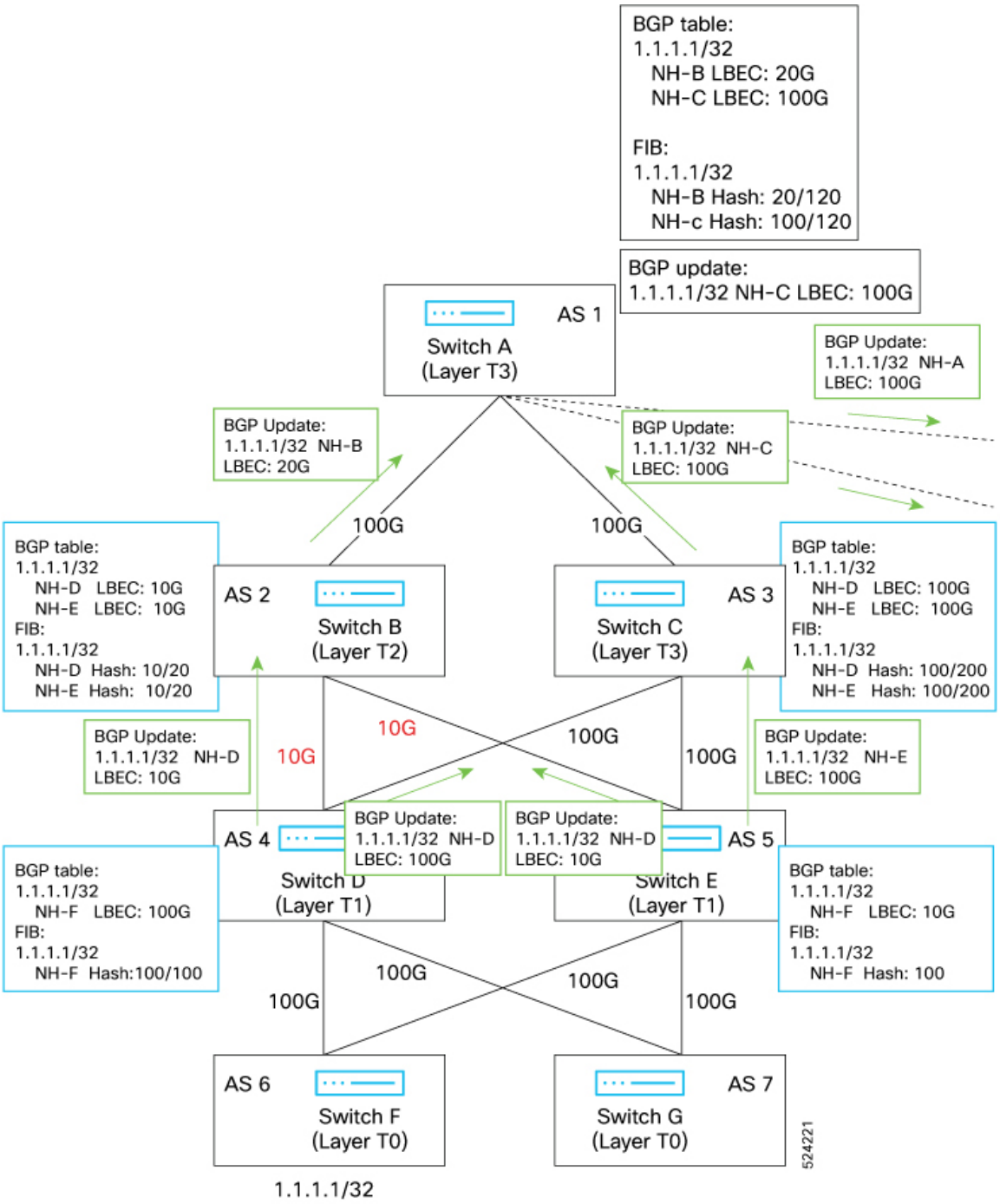

AS 1
Switch A
(Layer T3)
100G
100G
AS 2
Switch B
(Layer T2)
AS 3
Switch C
(Layer T3)
10G
10G
100G
100G
AS 4
Switch D
(Layer T1)
AS 5
Switch E
(Layer T1)
100G
100G
100G
100G
AS 6
Switch F
(Layer T0)
AS 7
Switch G
(Layer T0)
524220
1.1.1.1/32

図 *35* : 構成例 *2*



A から B および C への構成

```
router bgp 1
neighbor B
    address-family ipv4|v6 unicast
```

```
      link-bandwidth cumulative
      send-community extended
  neighbor C
    address-family ipv4|v6 unicast
      link-bandwidth cumulative
      send-community extended
  neighbor B'
    address-family ipv4|v6 unicast
      link-bandwidth cumulative
      send-community extended
  neighbor C'
    address-family ipv4|v6 unicast
      link-bandwidth cumulative
      send-community extended
```

B から D、A、および E への構成

```
router bgp 1
neighbor B
    address-family ipv4|v6 unicast
      link-bandwidth cumulative
      send-community extended
  neighbor C
    address-family ipv4|v6 unicast
      link-bandwidth cumulative
      send-community extended
  neighbor B'
    address-family ipv4|v6 unicast
      link-bandwidth cumulative
      send-community extended
  neighbor C'
    address-family ipv4|v6 unicast
      link-bandwidth cumulative
      send-community extended
```

C から A、D、および E への構成

```
router bgp 3
  neighbor A
    address-family ipv4|v6 unicast
      link-bandwidth cumulative
      send-community extended
  neighbor D
    address-family ipv4|v6 unicast
      link-bandwidth cumulative
      send-community extended
  neighbor E
    address-family ipv4|v6 unicast
      link-bandwidth cumulative
      send-community extended
```

D から F、B、および C への構成

```
router bgp 4
  neighbor F
    address-family ipv4|v6 unicast
      dmz-link-bandwidth
  neighbor B
    address-family ipv4|v6 unicast
      link-bandwidth cumulative
      send-community extended
neighbor C
```

```
    address-family ipv4|v6 unicast
       link-bandwidth cumulative
       send-community extended
```

E から F、B、および C への構成

```
router bgp 5
  neighbor F
    address-family ipv4|v6 unicast
      dmz-link-bandwidth
  neighbor B
    address-family ipv4|v6 unicast
       link-bandwidth cumulative
       send-community extended
neighbor C
    address-family ipv4|v6 unicast
       link-bandwidth cumulative
       send-community extended
```

設定の確認

構成を検証するには、次のコマンドを使用します：

- 次のコマンドを実行して、dmz-link-bandwidth コマンドがピア に対して有効になっているかどうかを確認します。

```
show bgp ipv4 unicast neighbors 192.168.11.2 | i i link
         dmz-link-bandwidth is enabled
```

- 次のコマンドを実行して、link-bandwidth cumulative コマンドがピアに対して有効になっているかどうかを確認します。

```
show bgp ipv4 unicast neighbors 10.1.1.2 | i i link
         link-bandwidth cumulative is enabled
```

- 次のコマンドを実行して、BGP パスにリンク帯域幅拡張コミュニティがあるかどうかを確認します。

```
show bgp ipv4 unicast 1.1.1.1/32
BGP routing table information for VRF default, address family IPv4 Unicast
BGP routing table entry for 1.1.1.1/32, version 403 Paths: (1 available, best #1) Flags:
 (0x8000001a) (high32 0x002000) on xmit-list, is in urib, is best urib route, is in HW

Advertised path-id 1 Path type: external, path is valid, is best path, no labeled
nexthop, in rib

AS-Path: 10 33299 51178 47751 {27016} , path sourced external to AS 192.168.11.2 (metric
 0) from 192.168.11.2 (192.168.11.2) Origin EGP, MED 2219, localpref 100, weight 0
Community: 1:1 Extcommunity: LB:1:125000000

Path type: external, path is valid, is multi-path, no labeled nexthop, in rib
AS-Path: 10 33299 51178 47751 {27016} , path sourced external to AS 192.168.11.3 (metric
 0) from 192.168.11.2 (192.168.11.2) Origin EGP, MED 2219, localpref 100, weight 0
Community: 1:1 Extcommunity: LB:1:250000000
```

- 次のコマンドを実行して、回送テーブルにハッシュ比率があるかどうかを確認します。

```
 show ip route 1.1.1.1/32 detail

100.1.1.1/32, ubest/mbest: 1/0 *via 192.168.11.2, [20/2219], 00:14:22, bgp-1, bw:333,
external, tag 10 client-specific data: 10 recursive next hop: 192.168.11.2/32 extended
```

```
 route information: BGP origin AS 0 BGP peer AS 10
100.1.1.2/32, ubest/mbest: 1/0 *via 192.168.11.3, [20/2219], 00:14:22, bgp-1, bw:666,
external, tag 10 client-specific data: 10 recursive next hop: 192.168.11.3/32 extended
 route information: BGP origin AS 0 BGP peer AS 10
```

デフォルト ルートのアドバタイズ

デフォルトのルート（ネットワーク 0.0.0.0）をアドバタイズするように BGP を設定できます。

始める前に

BGP をイネーブルにする必要があります（「BGP のイネーブル化」の項を参照）。

手順の概要

1. **configure terminal**
2. **route-map allow permit**
3. **exit**
4. **ip route** *ip-address network-mask* **null** *null-interface-number*
5. **router bgp** *as-number*
6. **address-family** {**ipv4** | **ipv6**} **unicast**
7. **default-information originate**
8. **redistribute static route-map allow**
9. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**route-map allow permit**
	例：
	`switch(config)# route-map allow permit`
	`switch(config-route-map)#`
		ルータのマップ コンフィギュレーション モードを開始し、ルートを再配布する条件を定義します。。

Step 3	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルータのマップ設定モードを終了します。

	コマンドまたはアクション	目的
Step 4	**ip route** *ip-address network-mask* **null** *null-interface-number* 例： `switch(config)# ip route 192.0.2.1 255.255.255.0 null 0`	IP アドレスを設定します。

Step 5	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		BGP モードを開始し、AS 番号をローカルの BGP スピーカに割り当てます。

Step 6	**address-family** {**ipv4** | **ipv6**} **unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		アドレスファミリ設定モードに入ります。

Step 7	**default-information originate** 例： `switch(config-router-af)# default-information originate`	デフォルトのルートをアドバタイズします。
Step 8	**redistribute static route-map allow** 例： `switch(config-router-af)# redistribute static route-map allow`	デフォルトのルートを再配布します。
Step 9	（任意） **copy running-config startup-config** 例： `switch(config-router-af)# copy running-config startup-config`	この設定変更を保存します。

BGP 属性フィルタリングの設定とエラー処理

Cisco NX-OS リリース 9.3(3) 以降では、BGP属性フィルタリングとエラー処理を設定して、セキュリティ レベルを向上させることができます。次の機能を利用でき、次の順序で実装されます。

- **パス属性 treat-as-withdraw:** アップデートに指定した属性タイプが含まれている場合に、指定したネイバーから受け取った BGP アップデートを treat-as-withdraw とすることを許可します。アップデートに含まれるプレフィックスは、ルーティング テーブルから削除されます。
- **パス属性 discard:**BGP アップデートの特定のパス属性を特定のネイバーから削除できます。
- **拡張属性エラー処理:** 形式が誤っているアップデートに起因するピア セッションのフラッピングを防止します。

属性タイプ 1、2、3、4、8、14、15、16 は、パス属性 treat-as-withdraw とパス属性 discard に対して設定できません。属性タイプ 9（Originator）、タイプ 10（Cluster-id）は、eBGP ネイバーでのみ設定できます。

BGP 更新メッセージからのパス属性の取り消しとしての処理

特定のパス属性を含むBGP更新を「扱うように」処理するには、ルータネイバーコンフィギュレーションモードで次のコマンドを使用します。

手順

	コマンドまたはアクション	目的

Step 1	[**no**] **path-attribute treat-as-withdraw** [*value* | **range** *start end*] **in**
	例：
	`switch#(config-router)# neighbor 10.20.30.40`
	`switch(config-router-neighbor)# path-attribute treat-as-withdraw 100 in`
	例：
	`switch#(config-router)# neighbor 10.20.30.40`
	`switch(config-router-neighbor)# path-attribute treat-as-withdraw range 21 255 in`
		指定されたパス属性またはパス属性の範囲を含む着信BGP更新メッセージをすべて取り消すものとして扱い、ルーティングテーブルが最新であることを確認するために着信ルートリフレッシュをトリガーします。treat-as-withdraw である BGP 更新のプレフィックスは、BGP ルーティング テーブルから削除されます。
		このコマンドは、BGPテンプレート ピアおよびBGPテンプレート ピア セッションでもサポートされます。

BGP 更新メッセージからのパス属性の破棄

特定のパス属性を含むBGPアップデートを廃棄するには、ルータ ネイバー コンフィギュレーション モードで次のコマンドを使用します。

手順

	コマンドまたはアクション	目的

Step 1	[**no**] **path-attribute discard** [*value* | **range** *start end*] **in**
	例：
	`switch#(config-router)# neighbor 10.20.30.40`
	`switch(config-router-neighbor)# path-attribute discard 100 in`
	例：
	`switch#(config-router)# neighbor 10.20.30.40`
	`switch(config-router-neighbor)# path-attribute discard range 100 255 in`
		指定されたネイバーの BGP アップデート メッセージ内の指定されたパス属性をドロップし、ルーティング テーブルが最新であることを確認するために着信ルート リフレッシュをトリガーします。特定の属性または不要な属性の範囲全体を設定できます。
		このコマンドは、BGPテンプレート ピアおよびBGPテンプレート ピア セッションでもサポートされます。
		（注）

	コマンドまたはアクション	目的
		discard と treat-as-withdaw の両方に同じパス属性が設定されている場合、treat-as-withdaw の優先順位が高くなります。

拡張属性エラー処理のイネーブル化またはディセーブル化

BGP 拡張属性エラー処理はデフォルトで有効になっていますが、無効にすることもできます。この機能は、RFC 7606 に準拠しており、不正な更新によるピアセッションのフラッピングを防止します。デフォルトの動作は、eBGP ピアと iBGP ピアの両方に適用されます。

拡張エラー処理を無効または再度有効にするには、ルータ設定モードで次のコマンドを使用します。

手順

	コマンドまたはアクション	目的

Step 1	[**no**] **enhanced-error**
	例：
	`switch(config)# router bgp 1000`
	`switch(config-router)# enhanced-error`
		BGP拡張属性エラー処理をいネーブルまたはディセーブルにします。

取り消されたパス属性または破棄されたパス属性の表示

廃棄または不明なパス属性に関する情報を表示するには、次のいずれかのタスクを実行します。

コマンド	目的
show bgp {**ipv4** | **ipv6**} **unicast path-attribute discard**]	属性が破棄されたすべてのプレフィックスを表示します。
show bgp {**ipv4** | **ipv6**} **unicast path-attribute unknown**]	不明な属性を持つすべてのプレフィックスを表示します。
show bgp {**ipv4** | **ipv6**} **unicast** *ip-address*	プレフィックスに関連付けられている不明な属性および破棄された属性を表示します。

次の例は、属性が廃棄されたプレフィックスを示しています。

```
switch# show bgp ipv4 unicast path-attribute discard
Network            Next Hop
1.1.1.1/32         20.1.1.1
1.1.1.2/32         20.1.1.1
1.1.1.3/32         20.1.1.1
```

次の例は、不明な属性を持つプレフィックスを示しています。

```
switch# show bgp ipv4 unicast path-attribute unknown
Network            Next Hop
2.2.2.2/32         20.1.1.1
2.2.2.3/32         20.1.1.1
```

次の例は、プレフィックスに関連付けられている不明な属性および破棄された属性を表示します。

```
switch# show bgp ipv4 unicast 2.2.2.2
BGP routing table entry for 2.2.2.2/32, version 6241
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
  1000
    20.1.1.1 from 20.1.1.1 (20.1.1.1)
      Origin IGP, localpref 100, valid, external, best
      unknown transitive attribute: flag 0xE0 type 0x62 length 0x64
        value 0000 0000 0100 0000 0200 0000 0300 0000
              0400 0000 0500 0000 0600 0000 0700 0000
              0800 0000 0900 0000 0A00 0000 0B00 0000
              0C00 0000 0D00 0000 0E00 0000 0F00 0000
              1000 0000 1100 0000 1200 0000 1300 0000
              1400 0000 1500 0000 1600 0000 1700 0000
              1800 0000
      rx pathid: 0, tx pathid: 0x0
      Updated on Jul 20 2019 07:50:43 PST
```

BGP の調整

一連のオプション パラメータを使用することによって、BGP 特性を調整できます。

BGP を調整するには、ルータ コンフィギュレーション モードで次のオプション コマンドを使用します。

コマンド	目的
bestpath [**always-compare-med** | **as-pathmultipath-relax** | **compare-routerid** |**cost-community ignore** | **igp-metric ignore** |**med** {**confed** |**missing-as-worst**| **non-deterministic**}] 例： `switch(config-router)# bestpath always-compare-med`	ベストパス アルゴリズムを変更します。オプション パラメータは次のとおりです。 • **always-compare-med**：異なる自律システム（AS）からのパスの MED を比較します。 • **as-path multipath-relax**： 異なる（ただし長さが等しい）AS パスを持つプロバイダ間でのロードシェアリングを許可します。このオプションを指定しないと、AS パスはロード シェアリングの場合に同一である必要があります。構成すると、BGP は異なる ASN からの場合でも、潜在的なマルチパスの中から MED が最も小さいベスト パスを選択します。 • **compare-routerid**：同一の eBGP パスのルータ ID を比較します。 • **cost-community ignore**：BGP ベストパス計算のコスト コミュニティを無視します。 • **igp-metric ignore**：ベスト パス選択時に内部ゲートウェイプロトコル（IGP）メトリックを無視します。このオプションは、Cisco NX-OS リリース 9.2(2) 以降で使用可能です。 • **med confed**：コンフェデレーション内からのパス間のみで MED を比較するように最適なパスを強制します。 • **med missing-as-worst**：消失 MED を最高の MED と見なします。 • **med non-deterministic**：同じ自律システムからのパスの中から最適な MED パスを決して選択しません。
enforce-first-as 例： `switch(config-router)# enforce-first-as`	ネイバー自律システムを eBGP の AS_path 属性で指定する最初の AS 番号にします。

コマンド	目的
log-neighbor-changes 例： `switch(config-router)# log-neighbor-changes`	ネイバーでステートが変化したときに、システム メッセージを生成します。 （注） 特定のネイバーのネイバー ステータス変化に関するメッセージを抑制するには、ルータ アドレス ファミリ コンフィギュレーション モードで **log-neighbor-changes disable** コマンドを使用できます。
router-id *id* 例： `switch(config-router)# router-id 10.165.20.1`	この BGP スピーカのルータ ID を手動で設定します。
timers [*prefix-peer-wait* | *bgp holdtime* | **prefix-peer-timeout** *timeout* | **bestpath-limit** *bestpath-timeout*] 例： `switch(config-router)# timers bestpath-limit 300`	BGP タイマー値を設定します。オプション パラメータは次のとおりです。 • *prefix-peer-wait* ：プレフィックス ピアの待機タイマー。有効な範囲は 0 ～ 1200 秒です。デフォルトは 90 です。 • *bgp* ：BGP セッション キープアライブ時間。有効な範囲は 0 ～ 3600 秒です。デフォルト値は 60 です。 • *holdtime* ：異なる bgp キープアライブとホールド時間。範囲は 0 ～ 3600 秒で、デフォルト値は 60 秒です。 • *timeout* ：プレフィックス ピア タイムアウト値。有効な範囲は 0 ～ 1200 秒です。デフォルト値は 30 です。 • *bestpath-timeout* ：ベストパス タイムアウトを秒単位で設定します。デフォルト値は 300 です。大規模な BGP セットアップが予想される場合、スケールに基づいて、タイムアウト値を 480～1200 に設定する必要があります。 このコマンドの設定後、BGP セッションを手動でリセットする必要があります。

BGP を調整するには、ルータ アドレス ファミリ設定モードで次のオプション コマンドを使用します。

コマンド	目的
distance *ebgp-distance ibgp-distance local-distance* 例： `switch(config-router-af)# distance 20 100 200`	BGP のアドミニストレーティブ ディスタンスを設定します。範囲は 1 ～ 255 です。デフォルトの設定は次のとおりです。 • *ebgp-distance* —20 • *ibgp-distance* —200 • *local-distance* —220 ローカル ディスタンスは、集約廃棄ルートが RIB に組み込まれている場合に、集約廃棄ルートに使用するアドミニストレーティブディスタンスです。 外部アドミニストレーティブディスタンスの値を入力したら、要件に応じて内部ルートのアドミニストレーティブディスタンスの値またはローカルルートのアドミニストレーティブディスタンスの値を入力する必要があります。内部/ローカルルートもルート管理で考慮されます。
log-neighbor-changes [**disable**] 例： `switch(config-router-af)# log-neighbor-changes disable`	この特定のネイバーの状態が変化すると、システム メッセージを生成します。 **disable** オプションを使用すると、この特定のネイバーのネイバー ステータス変化に関するメッセージが抑制されます。

BGP を調整するには、ネイバー コンフィギュレーション モードで次のオプション コマンドを使用します。

コマンド	目的
description *string* 例： `switch(config-router-neighbor)# description main site`	この BGP ピアを説明するストリングを設定します。ストリングには最大 80 の英数字を使用できます。
low-memory exempt 例： `switch(config-router-neighbor)# low-memory exempt`	メモリ不足状態によるシャットダウンからこの BGP ネイバーを除外します。
transport connection-mode passive 例： `switch(config-router-neighbor)# transport connection-mode passive`	受動接続の確立だけが可能です。この BGP スピーカは BGP ピアへの TCP 接続を開始しません。このコマンドの設定後、BGP セッションを手動でリセットする必要があります。

コマンド	目的
[**no** | **default**] **remove-private-as** [**all** |**replace-as**] 例： `switch(config-router-neighbor)# remove-private-as`	eBGP ピアへの発信ルートアップデートからプライベート AS 番号を削除します。このコマンドによって、BGP ネイバー セッションの自動ソフト クリアまたはリフレッシュが開始されます。 オプション パラメータは次のとおりです。 • **no**： コマンドをディセーブルにします。 • **default**： デフォルト モードにコマンドを移動します。 • **all**： AS パスからすべてのプライベート AS 番号を削除します。 • **replace-as**： すべてのプライベート AS 番号を replace-as AS-path 値に置き換えます。 このコマンドの詳細については、拡張 BGP に関する注意事項と制限事項（372 ページ）を参照してください。
update-source *interface-type number* 例： `switch(config-router-neighbor)# update-source ethernet 2/1`	ピアとの BGP セッション用に設定されたインターフェイスの送信元 IP アドレスを使用するように、BGP スピーカを設定します。このコマンドによって、BGP ネイバー セッションの自動通知およびセッション リセットが開始されます。単一ホップ iBGP ピアでは、**update-source** が設定されている場合に、高速外部フォールオーバーをサポートします。

BGP を調整するには、ネイバー アドレス ファミリ コンフィギュレーション モードで次のオプション コマンドを使用します。

コマンド	目的
allowas in 例： `switch(config-router-neighbor-af)# allowas in`	BRIP にインストールする AS パスにルート自体の AS を持つことを可能にします。
default-originate [**route-map** *map-name*] 例： `switch(config-router-neighbor-af)# default-originate`	BGP ピアへのデフォルト ルートを作成します。

コマンド	目的
disable-peer-as-check 例： `switch(config-router-neighbor-af)# disable-peer-as-check`	デバイスが同じ AS パスで一方のノードからもう一方のノードに学習されたルートをアドバタイズすると同時に、ピア AS 番号のチェックをディセーブルにします。
filter-list *list-name* {**in** | **out**} 例： `switch(config-router-neighbor-af)# filter-list BGPFilter in`	着信または発信ルートアップデートに関して、この BGP ピアに AS_path フィルタ リストを適用します。このコマンドによって、BGP ネイバー セッションの自動ソフト クリアまたはリフレッシュが開始されます。
prefix-list *list-name* {**in** | **out**} 例： `switch(config-router-neighbor-af)# prefix-list PrefixFilter in`	着信または発信ルートアップデートに関して、この BGP ピアにプレフィックス リストを適用します。このコマンドによって、BGP ネイバーセッションの自動ソフトクリアまたはリフレッシュが開始されます。
send-community 例： `switch(config-router-neighbor-af)# send-community`	この BGP ピアにコミュニティ属性を送信します。このコマンドによって、BGP ネイバーセッションの自動ソフトクリアまたはリフレッシュが開始されます。
send-community extended 例： `switch(config-router-neighbor-af)# send-community extended`	この BGP ピアに拡張コミュニティ属性を送信します。このコマンドによって、BGP ネイバーセッションの自動ソフトクリアまたはリフレッシュが開始されます。
suppress-inactive 例： `switch(config-router-neighbor-af)# suppress-inactive`	ベスト（アクティブ）ルートだけを BGP ピアにアドバタイズします。このコマンドによって、BGP ネイバー セッションの自動ソフト クリアまたはリフレッシュが開始されます。
[no | default] as-override 例： `switch(config-router-neighbor-af)# as-override`	**no -**（オプション）コマンドを無効にします。 **default :** （オプション）デフォルト モードにコマンドを移動します。 **as-override：** eBGP ピアに更新を送信する際に、パス属性内のピアの AS 番号をすべてローカル AS 番号に置き換えます。

ポリシーベースのアドミニストレーティブディスタンスの設定

設定されたルート マップで説明されているポリシーに一致する外部 BGP（eBGP）と内部 BGP（iBGP）の距離を設定できます。ルートマップで設定された距離は、一致するルートとともにユニキャスト RIB にダウンロードされます。BGP は最適パスを使用して、ユニキャスト RIB テーブルのネクストホップをダウンロードするときのアドミニストレーティブディスタンスを決定します。ポリシーに match 句または deny 句がない場合、BGP は distance コマンドで設定された距離またはルートのデフォルトの距離を使用します。

ポリシーベースのアドミニストレーティブ ディスタンス機能は、2 つの異なるルーティング プロトコルから同じ宛先に 2 つ以上のルートが存在する場合に役立ちます。

始める前に

BGP を有効にする必要があります。

手順の概要

1. switch# **configure terminal**
2. switch(config)# **ip prefix-list** *name* **seq** *number* **permit** *prefix-length*
3. switch(config)# **route-map** *map-tag* **permit** *sequence-number*
4. switch(config-route-map)# **match ip address prefix-list** *prefix-list-name*
5. switch(config-route-map)# **set distance** *value1 value2 value3*
6. switch(config-route-map)# **exit**
7. switch(config)# **router bgp** *as-number*
8. switch(config-router)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} **unicast**
9. switch(config-router-af)# **table-map** *map-name*
10. （任意） switch(config-router-af)# **show forwarding distribution**
11. （任意） switch(config)# **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	switch# **configure terminal**	グローバル コンフィギュレーション モードを開始します。
Step 2	switch(config)# **ip prefix-list** *name* **seq** *number* **permit** *prefix-length*	permit キーワードを使用して、IP パケットまたはルートを照合するためのプレフィクス リストを作成します。

	コマンドまたはアクション	目的
Step 3	switch(config)# **route-map** *map-tag* **permit** *sequence-number*	permit キーワードを使用してルート マップを作成し、ルート マップ コンフィギュレーション モードを開始します。ルートの一致基準がポリシー内で満たされると、パケットはポリシーでルーティングされます。
Step 4	switch(config-route-map)# **match ip address prefix-list** *prefix-list-name*	プレフィクス リストに基づいて IPv4 ネットワーク ルートを照合します。プレフィクスリスト名には最大 63 文字の英数字を使用できます。
Step 5	switch(config-route-map)# **set distance** *value1 value2 value3*	ローカル自律システムから発信される内部 BGP（iBGP）または外部 BGP （eBGP）ルートおよび BGP ルートのアドミニストレーティブ ディスタンスを指定します。範囲は 1 ～ 255 です。 外部アドミニストレーティブディスタンスの値を入力したら、要件に応じて内部ルートのアドミニストレーティブ ディスタンスの値またはローカル ルートのアドミニストレーティブディスタンスの値を入力する必要があります。内部/ローカルルートもルート管理で考慮されます。
Step 6	switch(config-route-map)# **exit**	ルート マップ設定モードを終了します。
Step 7	switch(config)# **router bgp** *as-number*	BGP モードを開始し、AS 番号をローカルの BGP スピーカに割り当てます。
Step 8	switch(config-router)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} **unicast**	アドレス ファミリ設定モードを開始します。
Step 9	switch(config-router-af)# **table-map** *map-name*	BGP ルートを RIB テーブルに転送する前にそのルートのルートマップの選択的アドミニストレーティブディスタンスを設定します。テーブルマップ名には最大 63 文字の英数字を使用できます。 （注） VRF アドレスファミリ設定モードで **table-map** コマンドを設定することもできます。
Step 10	（任意） switch(config-router-af)# **show forwarding distribution**	フォワーディング情報の配布を表示します。
Step 11	（任意） switch(config)# **copy running-config startup-config**	リブートおよびリスタート時に実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーして、変更を継続的に保存します。

マルチプロトコル BGP の設定

複数のアドレスファミリ（IPv4およびIPv6のユニキャストおよびマルチキャストルートを含む）をサポートするように MP-BGP を設定できます。

始める前に

BGPをイネーブルにする必要があります。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ip-address* **remote-as** *as-number*
4. **address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		BGPモードを開始し、ローカルBGPスピーカに自律システム番号を割り当てます。

Step 3	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65534`
	`switch(config-router-neighbor)#`
		BGPルーティング用のネイバー設定モードを開始し、ネイバー IP アドレスを設定します。

Step 4	**address-family** {**ipv4**	**ipv6**} {**unicast**	**multicast**}
			例：
			`switch(config-router-neighbor)# address-family ipv4 multicast`
			`switch(config-router-neighbor-af)#`
				アドレスファミリコンフィギュレーションモードを開始します。

	コマンドまたはアクション	目的
Step 5	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、ネイバーのマルチキャスト RPF に対して IPv4 および IPv6 ルートのアドバタイズおよび受信をイネーブルにする例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 2/1
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 multicast
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# address-family ipv6 multicast
switch(config-router-neighbor-af)# copy running-config startup-config
```

BMP の設定

Cisco NX-OS リリース 7.0(3)I5(2) 以降では、デバイスに BMP を設定できます。

始める前に

BGP をイネーブルにする必要があります（「BGP のイネーブル化」の項を参照）。

手順の概要

1. **configure terminal**
2. **router bgp as-number**
3. **bmp server** *server-number*
4. **address ip-address port-number port-number**
5. **description** *string*
6. **initial-refresh** { *skip* | *delay time*}
7. **initial-delay** *time*
8. **stats-reporting-period** *time*
9. shutdown
10. **vrf** *vrf-name*
11. **update-source** <*interface-name*>
12. **neighbor ip-address**
13. **remote-as** *as-number*
14. **bmp-activate-server** *server-number*

15. （任意） **show bgp bmp** *server [server-number] [detail]*
16. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例： `switch# configure terminal`	グローバル コンフィギュレーション モードを開始します。
Step 2	**router bgp as-number** 例： `switch(config)# router bgp 200`	BGP モードを開始し、ローカル BGP スピーカに自律システム番号を割り当てます。
Step 3	**bmp server** *server-number* 例： `switch(config-router-bmp)# bmp-server 1`	BGP が情報を送信する BMP サーバを設定します。サーバ番号がキーとして使用されます。 （注） 最大 2 つの BMP サーバを設定できます。
Step 4	**address ip-address port-number port-number** 例： `switch(config-router-bmp)# address 10.1.1.1 port-number 2000`	ホストの IPv4 または IPv6 アドレスと、BMP スピーカーが BMP サーバに接続するポート番号を設定します。
Step 5	**description** *string* 例： `switch(config-router-bmp)# description BMPserver1`	BMP サーバの説明を設定します。最大 256 文字の英数字を入力できます。
Step 6	**initial-refresh** { *skip* | *delay time*} 例： `switch(config-router-bmp)# initial-refresh delay 100`	BGP がコンバージされ、後で BMP サーバ接続が確立されたときにルートリフレッシュを送信するオプションを設定します。 skip オプションは、BMP サーバ接続が後でアップした場合にルートリフレッシュを送信しないことを指定します。 delay オプションは、ルート更新を送信するまでの時間を秒単位で指定します。有効範囲は 30 ～ 720 秒で、デフォルトは 30 秒です。
Step 7	**initial-delay** *time* 例： `switch(config-router-bmp)# initial-delay 120`	BMP サーバへの接続が試行されるまでの遅延を設定します。有効範囲は 30 ～ 720 秒で、デフォルトは 45 秒です。

	コマンドまたはアクション	目的
Step 8	**stats-reporting-period** *time* 例： `switch(config-router-bmp)# stats-reporting-period 50`	BMP サーバが BGP ネイバーから統計レポートを受信する時間間隔を設定します。有効範囲は30～720秒で、デフォルトはディスエーブルです。
Step 9	shutdown 例： `switch(config-router-bmp)# shutdown`	BMP サーバへの接続を無効にします。
Step 10	**vrf** *vrf-name* 例： `switch(config-router-bmp)# vrf BMP`	BMP サーバが到達可能な VRF を選択します。
Step 11	**update-source** <*interface-name*> 例： `switch(config-router-bmp)# update-source ethernet4/2`	BMP サーバ接続の確立に使用するローカルインターフェイスを選択します。
Step 12	**neighbor ip-address** 例： `switch(config-router-bmp)# neighbor 192.168.1.2`	BGP ルーティング用のネイバー コンフィギュレーション モードを開始し、ネイバー IP アドレスを設定します。
Step 13	**remote-as** *as-number* 例： `switch(config-router-neighbor)# remote-as 65535`	リモート BGP ピアの AS 番号を設定します。
Step 14	**bmp-activate-server** *server-number* 例： `switch(config-router-neighbor)# bmp-activate-server 1`	ネイバーの情報の送信先となる BMP サーバを設定します。
Step 15	（任意） **show bgp bmp** *server [server-number] [detail]* 例： `switch(config-router-neighbor)# show bgp bmp server`	BMP サーバ情報を表示します。
Step 16	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor)# copy running-config startup-config`	この設定変更を保存します。

BGP ローカル ルート リーク

BGP ローカル ルート リークについて

リリース 9.3(1) 以降、NX-OS BGP は、次の間のインポートされた VPN ルートのリークをサポートします。

- VPN ルート テーブルとデフォルト VRF ルート テーブル
- VPN ルート テーブルと VRF-Lite ルート テーブル
- リーフからリーフへの接続用のボーダー リーフ（BL）スイッチ ルート テーブル

この機能により、ルートテーブル間のルートの伝播が可能になります。インポートマップまたはエクスポートマップを設定することで、VRF のルートリークを制御できます。このマップには、ローカルで発生した着信ルートを許可または禁止し、アドバタイズするかどうかを指定するオプションが含まれています。ローカル ルート リークは双方向であるため、ローカルに発信されたルートは VRF から BGP VPN にリークされ、BGP VPN からインポートされたルートは VRF にリークされます。

（注） NX-OS は、中央集中型ルート リークと呼ばれる同様の機能をサポートしています。詳細については、「レイヤ 3 仮想化の設定 （535 ページ）」を参照してください。

BGP ローカル ルート リークの注意事項と制約事項

BGPローカルルートリーク機能の注意事項と制約事項は次のとおりです。

- この機能は、次のシスコ ハードウェアによりサポートされます。
  - この機能は、Cisco Nexus 9332C、9364C、9300-EX、9300-FX/FXP/FX2/FX3、および9300-GX プラットフォーム スイッチと、9700-EX/FX ラインカードを備えた Cisco Nexus 9500 プラットフォーム スイッチに導入されました。
  - -R ライン カード搭載の Cisco Nexus 9500 プラットフォーム スイッチ
- ルート ターゲットを使用する場合、同じルート ターゲットが同じリモート パスを指す重複パスを持っている可能性があり、これがスイッチのメモリとパフォーマンスに悪影響を及ぼす可能性があります。ルート ターゲットを使用する場合は注意してください。
- 同じ VRF 間で境界リーフ ルータ（BL）がリークするリーフツーリーフの場合に、ローカルルートリークを使用する場合は注意してください。このシナリオでは、ルーティングループが発生しやすくなります。インポートされたルートを他の BL から除外するには、インバウンド ルート マップを使用することを推奨します。

- リモート パスが取り消された後、BGP がパスを完全にクリーンアップするまでにさらに 20 秒かかることがあります。

デフォルト VRF にリークするために VPN からインポートされたルートを設定する

VRF を設定して、BGP VPN からインポートされたルートが、デフォルトの VRF へエクスポートされることを許可することができます。この手順は、デフォルト以外の VRF に使用します。

始める前に

BGP をまだ有効にしていない場合は、ここで有効にします（**feature bgp**）。

手順の概要

1. **config terminal**
2. **vrf context** *vrf-name*
3. **address-family** *address-family sub family*
4. **export vrf default [***prefix-limit***] map***route-map* **allow-vpn**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**config terminal**
	例：
	`switch-1# config terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**vrf context** *vrf-name*
	例：
	`switch-1(config)# vrf context vpn1`
	`switch-1(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。32 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

Step 3	**address-family** *address-family sub family*
	例：
	`switch-1(config-vrf)# address-family ipv4 unicast`
	`switch-1(config-vrf-af-ipv4)#`	

	コマンドまたはアクション	目的

Step 4	**export vrf default [***prefix-limit***] map***route-map* **allow-vpn**
	例：
	`switch-1(config-vrf-af-ipv4)# export vrf default map vpnmap1 allow-vpn`
	`switch-1(config-vrf-af-ipv4)#`
		現在の VRF を設定して、BGP VPN からインポートされたルートが、デフォルトの VRF へエクスポートされることを許可します。

デフォルト VRF からリークされたルートを VPN にエクスポートするための設定

デフォルト VRF からリークされたルートを BGP VPN にエクスポートできるように VRF を設定できます。この手順は、デフォルト以外の VRF に使用します。

始める前に

BGP をまだ有効にしていない場合は、ここで有効にします（**feature bgp**）。

手順の概要

1. **config terminal**
2. **vrf context** *vrf-name*
3. **address-family** *address-family sub family*
4. **import vrf default [***prefix-limt***] map***route-map* **advertise-vpn**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**config terminal**
	例：
	`switch-1# config terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**vrf context** *vrf-name*
	例：
	`switch-1(config)# vrf context vpn1`
	`switch-1(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。32 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

Step 3	**address-family** *address-family sub family* 例：	

	コマンドまたはアクション	目的
	`switch-1(config-vrf)# address-family ipv4 unicast` `switch-1(config-vrf-af-ipv4)#`	

Step 4	**import vrf default [***prefix-limt***] map***route-map* **advertise-vpn**
	例：
	`switch-1(config-vrf-af-ipv4)# import vrf map vpnmap1 advertise-vpn`
	`switch-1(config-vrf-af-ipv4)#`
		デフォルト VRF からインポートされたルートを BGP VPN にエクスポートできるように現在の VRF を設定します。

VRF にエクスポートするために VPN からインポートしたルートの設定

VPN でインポートされたルートを別の VRF にエクスポートできるように VRF を設定できます。この手順は、デフォルト以外の VRF に使用してください。

始める前に

BGP をまだ有効にしていない場合は、ここで有効にします（**feature bgp**）。

手順の概要

1. **config terminal**
2. **vrf context** *vrf-name*
3. **address-family** *address-family sub family*
4. **export vrf allow-vpn**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**config terminal**
	例：
	`switch-1# config terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**vrf context** *vrf-name*
	例：
	`switch-1(config)# vrf context vpn1`
	`switch-1(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。32 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

Step 3	**address-family** *address-family sub family* 例：	

	コマンドまたはアクション	目的
	`switch-1(config-vrf)# address-family ipv4 unicast` `switch-1(config-vrf-af-ipv4)#`	

Step 4	**export vrf allow-vpn**
	例：
	`switch-1(config-vrf-af-ipv4)# export vrf allow-vpn`
	`nxosv2(config-vrf-af-ipv4)#`
		BGP VPM からインポートしたルートをデフォルト以外の VRF にエクスポートできるように VRF を設定します。

VRF からインポートして VPN にエクスポートするルートの設定

VRF は、別の VRF からインポートされたルートを BGP VPN にエクスポートできるように設定することができます。この手順は、デフォルト以外の VRF に使用してください。

始める前に

BGP をまだ有効にしていない場合は、ここで有効にします（**feature bgp**）。

手順の概要

1. **config terminal**
2. **vrf context** *vrf-name*
3. **address-family** *address-family sub family*
4. **import vrf advertise-vpn**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**config terminal**
	例：
	`switch-1# config terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**vrf context** *vrf-name*
	例：
	`switch-1(config)# vrf context vpn1`
	`switch-1(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。32 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

Step 3	**address-family** *address-family sub family*
	例：
	`switch-1(config-vrf)# address-family ipv4 unicast`
	`switch-1(config-vrf-af-ipv4)#`	

	コマンドまたはアクション	目的

Step 4	**import vrf advertise-vpn**
	例：
	`switch-1(config-vrf-af-ipv4)# import vrf advertise-vpn`
	`nxosv2(config-vrf-af-ipv4)#`
		別の VRF からインポートされたルートを BGP VPN にエクスポートできるように現在の VRF を設定します。

設定例

次に、BGP ローカル ルート リーク機能の設定例を示します。

BGP VPN からデフォルト VPN への到達可能性の設定

この例では、VPN とデフォルト VRF の間にある、VRF_A と呼ばれる中間 VRF を介して、ルートの再インポートを有効にします。

```
vrf context VRF_A
   address-family ipv4 unicast
   route-target both auto evpn
   import vrf default map MAP_1 advertise-vpn
   export vrf default map MAP_1 allow-vpn
```

ルートの再インポートは、VPN から VRF_A へのルートのインポートを制御する **advertise-vpn** オプションを使用して、また、VRF_A からデフォルト VRF への VPN インポート ルートのエクスポートを制御する、エクスポート マップのための **allow-vpn** を使用して有効にできます。設定は中間 VRF で行われます。

VPN から VRF-Lite への到達可能性の設定

この例では、VPN は VRF_A と呼ばれるテナント VRF に接続します。VRF_A は、VRF-B と呼ばれる VRF-Lite に接続します。この設定により、VPN でインポートされたルートを VRF_A から VRF_B にリークできます。

```
vrf context VRF_A
   address-family ipv4 unicast
   route-target both auto
   route-target both auto evpn
   route-target import 3:3
   route-target export 2:2
   import vrf advertise-vpn
   export vrf allow-vpn
vrf context VRF_B
   address-family ipv4 unicast
   route-target both 1:1
   route-target import 2:2
   route-target export 3:3
```

2 つの間のルート リークは、VRF_A（テナント）で設定されたエクスポート マップで **allow-vpn** を使用してイネーブルにします。VRF_A のエクスポート マップでは、VPN からインポートされたルートを VRF_B にリークできます。エクスポート マップによって処理されたルートは、ルート ターゲットのルート セットに追加される、**route-mapexport** および **export-map** 属性を持ちま

す。インポート マップは、**advertise-vpn** を使用して、VRF-Lite からインポートされたルートを VPN にエクスポートできるようにします。

VRF 間でルート リークが発生すると、ルートは再発信され、そのルート ターゲットは、新しい VRF の設定で指定されたルート ターゲット エクスポートおよびエクスポート マップ属性で置き換えられます。

リーフからリーフへの到達可能性

この例では、2 つの VPN と 2 つの VRF が存在します。VPN_1 は VRF_A に接続され、VPN_2 は VRF_B に接続されます。両方の VRF はルート識別子（RD）です。

```
vrf context VRF_A
   address-family ipv4 unicast
   route-target both auto
   route-target both auto evpn
   route-target import 3:3
   route-target export 2:2
   import vrf advertise-vpn
   export vrf allow-vpn
vrf context VRF_B
   address-family ipv4 unicast
   route-target both 1:1
   route-target import 2:2
   route-target export 3:3
   import vrf advertise-vpn
   export vrf allow-vpn
```

この 2 つの間のルート リークは、VRF_A および VRF_B で設定されたエクスポート マップの **allow-vpn** で有効にされます。VPN によってインポートされたルートには、ルート ターゲットのルート セットに追加された **route-mapexport** と **export-map** 属性があります。インポート マップのマップは、各 VRF からインポートされたルートが VPN にエクスポートされるようにする advertise-vpn オプションを使用します。

VRF 間でルート リークが発生すると、ルートは再発信され、そのルート ターゲットは、新しい VRF の設定で指定されたルート ターゲット エクスポートおよびエクスポート マップ属性で置き換えられます。

ループ防止付きリーフツーリーフ

リーフツーリーフ設定では、ルートマップに注意を払わないでいると、同じVRF間でリークしている BL 間のループが誤って発生する可能性があります。

- 各 BL でインバウンド ルート マップを使用すれば、他のすべての BL からの更新を拒否できます。
- BL がルートを発信する場合には、標準コミュニティを適用できます。これにより、他の BL はルートを受け入れることができます。このコミュニティは、受信側のBLで削除されます。

次の例では、VTEP 3.3.3.3、4.4.4.4、および 5.5.5.5 が BL です。

```
ip prefix-list BL_PREFIX_LIST seq 5 permit 3.3.3.3/32
ip prefix-list BL_PREFIX_LIST seq 10 permit 4.4.4.4/32
ip prefix-list BL_PREFIX_LIST seq 20 permit 5.5.5.5/32
ip community-list standard BL_COMMUNITY seq 10 permit 123:123
route-map INBOUND_MAP permit 5
```

```
  match community BL_COMMUNITY
  set community none
route-map INBOUND_MAP deny 10
  match ip next-hop prefix-list BL_PREFIX_LIST
route-map INBOUND_MAP permit 20
route-map OUTBOUND_SET_COMM permit 10
  match evpn route-type 2 mac-ip
  set community 123:123
route-map SET_COMM permit 10
  set community 123:123
route-map allow permit 10

vrf context vni100
  vni 100
  address-family ipv4 unicast
    route-target import 2:2
    route-target export 1:1
    route-target both auto
    route-target both auto evpn
    import vrf advertise-vpn
    export vrf allow-vpn

vrf context vni200
  vni 200
  address-family ipv4 unicast
    route-target import 1:1
    route-target export 2:2
    route-target both auto
    route-target both auto evpn
    import vrf advertise-vpn
    export vrf allow-vpn

router bgp 100
  template peer rr
    remote-as 100
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
      route-map INBOUND_MAP in
      route-map OUTBOUND_SET_COMM out
  neighbor 101.101.101.101
    inherit peer rr
  neighbor 102.102.102.102
    inherit peer rr
  vrf vni100
    address-family ipv4 unicast
      network 3.3.3.100/32 route-map SET_COMM
  vrf vni200
    address-family ipv4 unicast
      network 3.3.3.200/32 route-map SET_COMM
```

この例では、ボーダーリーフ（BL）ルータのテナント VRF は追加のインポート エクスポート フローを有効にすることで、トラフィックをリークできます。ルートマップ内のルートターゲットは、ルートのインポート元またはエクスポート先を決定します。

VRF のマルチパス

この例では、VPN に複数の着信パスがあります。この設定により、VRF_A と呼ばれる中間 VRF（VPN と別の VRF の間にあり、VRF_B と呼ばれるもの）を介したルート リークが可能になります。マルチパスが VRF_A で有効になっているとします。

```
vrf context VRF_A
   address-family ipv4 unicast
   route-target both auto evpn
   route-target export 3:3
   export vrf allow-vpn
vrf context VRF_B
   address-family ipv4 unicast
   route-target import 3:3
```

ルートリークは、VRF_A で設定されたエクスポート マップの **allow-vpn** で有効になっています。特定のプレフィックスの 2 つのパスが VPN から学習されて VRF_A にインポートされると、同じ送信元 RD（VRF_A のローカル RD）を持つ 2 つの異なるパスが VRF_B に存在するようになります。各ルートは、元の送信元 RD（リモート RD）によって区別されます。

パスの重複

この例では、設定により単一の VPN パスを VRF_A と VRF_B の両方にインポートできるようになっています。VRF_A は **export vrf allow-vpn** で設定されているため、VRF_A もそのルートを VRF_B にリークします。VRF_B には同じ送信元 RD（VRF_A のローカル RD）を持つ 2 つのパスがありますが、それらは元の送信元 RD（リモート RD）によって区別されます。

```
vrf context VRF_A
   address-family ipv4 unicast
      route-target import 1:1 evpn
      route-target export 1:1 evpn
      route-target export 2:2
      export vrf allow-vpn
vrf context VRF_B
   address-family ipv4 unicast
     route-target import 1:1 evpn
     route-target import 2:2
```

この設定では、マルチパスが存在しない状況が発生します。

BGP ローカル ルート リーク情報の表示

次の show コマンドには、BGP ローカル ルート リーク機能に関する情報が含まれています。

コマンド	アクション
show bgp vrf *vrf-name* **process**	デフォルトまたはデフォルト以外のVRFの場合、**import advertise-vpn** および **export allow-vpn** オプションのイネーブル状態（YesまたはNo）が表示されます。
show bgp vrf *vrf-name* **ipv4 unicast** *prefix*	ルートのインポート元の宛先のリストなど、インポートされたパスに関する情報を表示します。

BGP グレースフル シャットダウン

BGP グレース フル シャットダウンに関する情報

リリース 9.3(1) 以降、BGP はグレースフル シャットダウン機能をサポートしています。この BGP 機能は、BGP **shutdown** コマンドと連携して次のことを行います。

- ルータまたはリンクがオフラインになったときのネットワーク コンバージェンス時間を大幅に短縮します。
- ルータまたはリンクがオフラインになったときに、転送中のドロップされたパケットを削減または排除します。

名前にかかわらず、BGP グレースフル シャットダウンは実際にはシャットダウンを引き起こしません。代わりに、ルータまたはリンクが間もなくダウンすることを、接続されているルータに通知します。

グレースフル シャットダウン機能は、GRACEFUL_SHUTDOWN ウェルノウン コミュニティ（0xFFFF0000 または 65535:0）を使用します。これは、IANA および IETF によって RFC 8326 によって識別されます。この既知のコミュニティは任意のルートにアタッチでき、ルートの他の属性と同様に処理されます。

この機能は、ルータまたはリンクがダウンすることを通知するため、メンテナンス時間帯または計画停止の準備に役立ちます。トラフィックへの影響を制限するには、BGP をシャットダウンする前にこの機能を使用します。

グレースフル シャットダウンの認識とアクティブ化

BGPルータは、すべてのルートの優先事項を、GRACEFUL SHUTDOWN 対応というコンセプトを通し、GRACEFUL_SHUTDOWN コミュニティによって制御できます。グレースフル シャットダウン対応は、デフォルトでイネーブルになっています。これにより、受信側ピアは、GRACEFUL_SHUTDOWN コミュニティを伝える着信ルートを優先しなくなります。一般的な使用例ではありませんが、**graceful-shutdown aware** コマンドを使用して、グレースフル シャットダウン対応を無効にしてから再度有効にすることもできます。

グレースフル シャットダウン対応は、BGP グローバル コンテキストでのみ適用されます。コンテキストの詳細については、グレースフル シャットダウンのコンテキスト（465 ページ）を参照してください。対応のためのオプションは、**activate** という別のオプションと一緒に動作します。このオプションをルートマップに割り当てると、グレースフル シャットダウンのルートをより詳細に制御できます。

グレースフル シャットダウン対応オプションとアクティブ化オプションの協同作用

グレースフル シャットダウンがアクティブな場合、**activate** キーワードを指定した場合にのみ、GRACEFUL_SHUTDOWN コミュニティがルート更新に追加されます。この時点で、コミュニティを含む新しいルート更新が生成され、送信されます。**graceful-shutdown aware** コマンドが設定さ

れると、コミュニティを受信するすべてのルータは、アップデート内のルートの優先を解除します（そのルート優先度を下げます）。**graceful-shutdown aware** コマンドを使用しなかった場合、BGPはGRACEFUL_SHUTDOWN コミュニティの設定されたルートの優先度を下げません。

この機能がアクティブになり、ルータがグレースフル シャットダウンの対応状態になった場合でも、BGP は引き続き、GRACEFUL_SHUTDOWN コミュニティが有効だとしてルートを考慮します。ただし、これらのルートには、最適パスの計算で最低の優先度が与えられます。代替パスが使用可能な場合は、新しい最適パスが選択され、まもなくダウンするルータまたはリンクに対応するためのコンバージェンスが行われます。

グレースフル シャットダウンのコンテキスト

BGPのグレースフル シャットダウン機能には、機能の影響と使用可能な機能を決定する 2 つのコンテキストがあります。

コンテキスト	影響	コマンド
グローバル	スイッチ全体と、スイッチによって処理されるすべてのルート。たとえば、GRACEFUL_SHUTDOWN コミュニティを持つすべてのルートを再アドバタイズします。	**graceful-shutdown activate [route-map** ルート マップ**]** **graceful-shutdown aware**
Peer	BGP ピアまたはネイバー間のリンク。たとえば、ピア間のリンクを 1 つだけ GRACEFUL_SHUTDOWNコミュニティでアドバタイズします。	**graceful-shutdown activate [route-map** ルート マップ**]**

ルート マップによるグレースフル シャットダウン

グレースフル シャットダウンは、ルート ポリシー マネージャ（RPM）機能と連携して、スイッチの BGP ルータが GRACEFUL_SHUTDOWN コミュニティを使用してルートを送受信する方法を制御します。ルート マップは、インバウンドおよびアウトバウンド方向でコミュニティとのルート更新を処理できます。通常、ルート マップは必要ありません。ただし、必要に応じて、グレースフル シャットダウン ルートの制御をカスタマイズするために使用できます。

通常のインバウンド ルート マップ

通常のインバウンド ルート マップは、BGP ルータに着信するルートに影響します。ルータはデフォルトでグレースフル シャットダウンを認識するため、通常のインバウンド ルート マップはグレースフル シャットダウン機能では一般的に使用されません。

Cisco NX-OS リリース 9.3 (1) 以降を実行している Cisco Nexus スイッチでは、グレースフル シャットダウン機能のインバウンド ルート マップは必要ありません。Cisco NX-OSリリース9.3（1）以降には、BGPルータがグレースフルシャットダウン対応である場合にGRACEFUL_SHUTDOWNコミュニティを持つすべてのルートを自動的に非優先にする、暗黙のインバウンドルート マップがあります。

通常のインバウンド ルート マップは、既知の GRACEFUL_SHUTDOWN コミュニティと一致するように設定できます。これらの着信ルート マップは一般的ではありませんが、使用される場合があります。

- スイッチが 9.3 (1) よりも前の Cisco NX-OS リリースを実行している場合、NX-OS 9.3 (1) には暗黙的なインバウンドルートマップがありません。これらのスイッチでグレースフル シャットダウン機能を使用するには、グレースフル シャットダウン インバウンド ルート マップを作成する必要があります。ルート マップは、既知の GRACEFUL_SHUTDOWN コミュニティを持つインバウンド ルートと一致し、それらを許可し、それらを非優先にする必要があります。着信ルート マップが必要な場合は、9.3 (1) より前のバージョンの NX-OS を実行し、グレースフル シャットダウン ルートを受信しているBGPピアで作成します。
- グレースフル シャットダウン認識をディセーブルにし、一部の BGP ネイバーからの GRACEFUL_SHUTDOWN コミュニティを持つ着信ルートでルータを動作させる場合は、それぞれのピアでインバウンド ルート マップを設定できます。

通常のアウトバウンド ルート マップ

通常のアウトバウンドルートマップは、BGPルータが送信するルートの転送を制御します。通常のアウトバウンド ルート マップは、グレースフル シャットダウン機能に影響を与える可能性があります。たとえば、GRACEFUL_SHUTDOWN コミュニティで一致するようにアウトバウンド ルート マップを設定し、属性を設定できます。これは、グレースフル シャットダウン アウトバウンド ルート マップよりも優先されます。

グレースフル シャットダウン アウトバウンド ルート マップ

アウトバウンド グレースフル シャットダウン ルート マップは、グレースフル シャットダウン機能のアウトバウンドルートマップの特定のタイプです。これらはオプションですが、ルートマップに関連付けられているコミュニティ リストがすでにある場合に役立ちます。通常のグレースフル シャットダウン アウトバウンド ルート マップには、特定の属性を設定または変更するための `set` 句のみが含まれています。

アウトバウンド ルート マップは、次の方法で使用できます。

- 既存のアウトバウンドルートマップをすでに持っている顧客の場合は、より大きいシーケンス番号を持つ新しいエントリを追加し、GRACEFUL_SHUTDOWN ウェルノウンコミュニティで照合し、必要な属性を追加できます。
- **graceful-shutdown activate route-map** *name* オプションを使用してグレースフル シャットダウン アウトバウンドルート マップを使用することもできます。これが一般的な使用例です。

  このルート マップには match 句が必要ないため、ルート マップはネイバーに送信されるすべてのルートで一致します。

ルート マップの優先順位

同じルータ上に複数のルート マップが存在する場合は、次の優先順位が適用されて、コミュニティとのルートの処理方法が決定されます。次の例を考慮してください。60 のローカル設定を設定する標準の発信ルート マップ名 Red があるとします。また、Blue という名前のピア グレースフル シャットダウン ルート マップがあり、local-pref が 30 に設定されているとします。ルート更新が処理されると、Red は Blue を上書きするため、ローカル プリファレンスは 60 に設定されます。

- 通常の発信ルートマップは、ピア グレースフル シャットダウンマップよりも優先されます。
- ピア グレースフル シャットダウン マップは、グローバル グレースフル シャットダウン マップよりも優先されます。

注意事項と制約事項

BGP グローバル シャットダウンの制限事項と注意事項は、次のとおりです。

- グレースフル シャットダウン機能は、影響を受けるルータの代替ルートがネットワークに存在する場合にのみ、トラフィック損失を回避するのに役立ちます。ルータに代替ルートがない場合は、GRACEFUL_SHUTDOWN コミュニティを伝送するルートが使用可能な唯一のルートであるため、最適パスの計算に使用されます。この状況では、機能の目的が失われます。
- GRACEFUL_SHUTDOWN コミュニティを送信するには、BGP 送信コミュニティの設定が必要です。
- ルート マップの場合:
  - グローバル ルート マップとネイバー ルート マップが設定されている場合、ネイバー単位のルート マップが優先されます。
  - 発信ルート マップは、グレースフル シャットダウン用に設定されたグローバル ルート マップよりも優先されます。
  - 発信ルート マップは、グレースフル シャットダウン用に設定されたピア ルート マップよりも優先されます。
  - レガシー（既存の）インバウンド ルート マップにグレースフル シャットダウン機能を追加するには、次の手順を実行します。
    1. graceful shutdown match 句をルート マップの先頭に追加します。これには、句に低いシーケンス番号（たとえば、シーケンス番号 0）を設定します。
    2. graceful shutdown 句の後に continue ステートメントを追加します。continue ステートメントを省略すると、graceful shutdown 句と一致するルートマップ処理が停止します。シーケンス番号が大きい他の句（たとえば、1 以上）は処理されません。

グレースフル シャットダウン タスクの概要

グレースフル シャットダウン機能を使用するには、通常、すべての Cisco Nexus スイッチでグレースフル シャットダウン対応をイネーブルにし、機能をイネーブルのままにします。BGP ルータをオフラインにする必要がある場合は、graceful-shutdown activate を設定します。

次の詳細に、グレースフル シャットダウン機能を使用するためのベスト プラクティスを示します。

ルータまたはリンクをダウンさせるには、次の手順を実行します。

1. グレースフル シャットダウン機能を設定します。
2. ネイバーでベスト パスを確認します。
3. 最適パスが再計算されたら、BGP を無効にする **shutdown** コマンドを発行します。
4. ルータまたはリンクをシャットダウンする必要がある作業を実行します。

ルータまたはリンクをオンラインに戻すには、次の手順を実行します。

1. シャットダウンが必要な作業が完了したら、BGP を再度イネーブルにします（**no shutdown**）。
2. グレースフル シャットダウン機能を無効にします（config モードの **no graceful-shutdown activate**）。

リンクのグレースフル シャットダウンの設定

この作業では、2 つの BGP ルータ間の特定のリンクでグレースフル シャットダウンを設定できます。

始める前に

BGP をまだ有効にしていない場合は、ここで有効にします（**feature bgp**）。

手順の概要

1. **config terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor {** *ipv4-address*|*ipv6-address* **} remote-as** *as-number*
4. **graceful-shutdown activate [route-map** *map-name***]**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**config terminal**
	例：
	`switch-1# configure terminal`
	`switch-1(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router bgp** *autonomous-system-number*
	例：
	`switch-1(config)# router bgp 110`
	`switch-1(config-router)#`
		ルータ コンフィギュレーション モードを開始して、BGP ルーティング プロセスを作成または設定します。

Step 3	**neighbor {** *ipv4-address*|*ipv6-address* **} remote-as** *as-number*
	例：
	`switch-1(config-router)# neighbor 10.0.0.3 remote-as 200`
	`switch-1(config-router-neighbor)#`
		ネイバーが属する自律システム（AS）を設定します。

Step 4	**graceful-shutdown activate [route-map** *map-name***]**
	例：
	`switch-1(config-router-neighbor)# graceful-shutdown activate route-map gshutPeer`
	`switch-1(config-router-neighbor)#`
		ネイバーへのリンクでグレースフル シャットダウンを設定します。また、既知の GRACEFUL_SHUTDOWN コミュニティを使用してルートをアドバタイズし、アウトバウンド ルート更新にルートマップを適用します。
		ルートは、デフォルトでグレースフル シャットダウン コミュニティでアドバタイズされます。この例では、ルートは gshutPeer という名前のルート マップを使用して、グレースフル シャットダウン コミュニティを持つネイバーにアドバタイズされます。
		gshut コミュニティを受信したデバイスは、ルートのコミュニティを確認し、オプションでコミュニティを使用してルーティング ポリシーを適用します。

GRACEFUL_SHUTDOWN コミュニティに基づく BGP ルートのフィルタリングとローカルプリファレンスの設定

まだ 9.3(1) を実行していないスイッチには、GRACEFUL_SHUTDOWN コミュニティ名と一致するインバウンド ルートマップがありません。したがって、正しいルートを識別して先送りする方法はありません。

9.3(1) よりも前のリリースの NX-OS を実行しているスイッチでは、グレースフル シャットダウン（65535:0）のコミュニティ値と一致するインバウンド ルート マップを設定し、ルートを非優先にする必要があります。

スイッチが 9.3(1) 以降を実行している場合、着信ルートマップを設定する必要はありません。

手順の概要

1. **configure terminal**
2. **ip community list standard** *community-list-name* **seq** *sequence-number* **{ permit | deny }** *value*
3. **route map** *map-tag* **{deny | permit}** *sequence-number*
4. **match community** *community-list-name*
5. **set local-preference** *local-pref-value*
6. **exit**
7. **router bgp** *community-list-name*
8. **neighbor {** *ipv4-address|ipv6-address* **}**
9. **address-family {** *address-family sub family* **}**
10. **send community**
11. **route map** *map-tag* **in**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch-1# configure terminal`
	`switch-1<config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**ip community list standard** *community-list-name* **seq** *sequence-number* **{ permit | deny }** *value*
	例：
	`switch-1(config)# ip community-list standard GSHUT seq 10 permit 65535:0`
	`switch-1(config)#`
		コミュニティ リストを設定し、よく知られたグレースフル シャットダウン コミュニティ値を持つルートを許可または拒否します。

Step 3	**route map** *map-tag* **{deny | permit}** *sequence-number*
	例：
	`switch-1(config)# route-map RM_GSHUT permit 10`
	`switch-1(config-route-map)#`
		ルート マップをシーケンス 10 として設定し、GRACEFUL_SHUTDOWN コミュニティを持つルートを許可します。

Step 4	**match community** *community-list-name*
	例：
	`switch-1(config-route-map)# match community GSHUT`
	`switch-1(config-route-map)#`
		IP コミュニティ リスト GSHUT に一致するルートがルート ポリシー マネージャ（RPM）により処理されるように設定します。

	コマンドまたはアクション	目的

Step 5	**set local-preference** *local-pref-value*
	例：
	`switch-1(config-route-map)# set local-preference 10`
	`switch-1(config-route-map)#`
		IP コミュニティ リスト GSHUT に一致するルートに、指定されたローカルプリファレンスが与えられるように設定します。

Step 6	**exit**
	例：
	`switch-1(config-route-map)# exit`
	`switch-1(config)#`
		ルートマップ設定モードを終了し、グローバル設定モードに戻ります。

Step 7	**router bgp** *community-list-name*
	例：
	`switch-1(config)# router bgp 100`
	`switch-1(config-router)#`
		ルータ設定モードを開始し、BGPインスタンスを作成します。

Step 8	**neighbor {** *ipv4-address|ipv6-address* **}**
	例：
	`switch-1(config-router)# neighbor 10.0.0.3`
	`switch-1(config-router-neighbor)#`
		指定したネイバーのルート BGP ネイバー モードを開始します。

Step 9	**address-family {** *address-family sub family* **}**
	例：
	`nxosv2(config-router-neighbor)# address-family ipv4 unicast`
	`nxosv2(config-router-neighbor-af)#`
		ネイバーをアドレス ファミリ（AF）設定モードにします。

Step 10	**send community**
	例：
	`nxosv2(config-router-neighbor-af)# send-community`
	`nxosv2(config-router-neighbor-af)#`
		ネイバーとの BGP コミュニティ交換を可能にします。

Step 11	**route map** *map-tag* **in**
	例：
	`nxosv2(config-router-neighbor-af)# route-map RM_GSHUT in`
	`nxosv2(config-router-neighbor-af)#`
		ネイバーからの着信ルートにルートマップを適用します。この例では、RM_GSHUT という名前のルート マップは、ネイバーからの GRACEFUL_SHUTDOWN コミュニティを持つルートを許可します。

すべての BGP ネイバーのグレースフル シャットダウンの設定

グレースフル シャットダウン イニシエータのすべてのネイバーに GRACEFUL_SHUTDOWN ウェルノウン コミュニティを手動で適用できます。

すべての BGP ネイバーに対して、グローバル レベルでグレースフル シャットダウンを設定できます。

始める前に

BGP をまだ有効にしていない場合は、ここで有効にします（**feature bgp**）。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **graceful-shutdown activate [route-map** *map-name***]**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch-1# configure terminal`
	`switch-1(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router bgp** *autonomous-system-number*
	例：
	`switch-1(config)# router bgp 110`
	`switch-1(config-router)#`
		ルータ コンフィギュレーション モードを開始して、BGP ルーティング プロセスを作成または設定します。

Step 3	**graceful-shutdown activate [route-map** *map-name***]**
	例：
	`switch-1(config-router-neighbor)# graceful-shutdown activate route-map gshutPeer`
	`switch-1(config-router-neighbor)#`
		すべてのネイバーへのリンクのグレースフル シャットダウン ルートマップを設定します。また、既知の GRACEFUL_SHUTDOWN コミュニティを持つすべてのルートをアドバタイズし、ルートマップをアウトバウンド ルート アップデートに適用します。
		ルートはデフォルトで GRACEFUL_SHUTDOWN コミュニティでアドバタイズされます。この例では、ルートが gshutPeer という名前のルートマップを持つコミュニティを持つすべてのネイバーにアドバタイズされます。ルートマップには set 句のみを含める必要があります。
		GRACEFUL_SHUTDOWN コミュニティを受信したデバイスは、ルートのコミュニティを確認し、オプションでコミュニティを使用してルーティング ポリシーを適用します。

GRACEFUL_SHUTDOWNコミュニティを使用したすべてのルートのプリファレンスの制御

Cisco NX-OS では、GRACEFUL_SHUTDOWN コミュニティを持つ着信ルートの優先順位を下げることができます。**graceful shutdown aware** が有効になっている場合、最適パス計算時に、BGP はコミュニティを伝送するルートを最も低い優先順位と見なします。デフォルトでは、プレファレンスの引き下げが有効になっていますが、このオプションを選択的に無効にすることもできます。

このオプションをイネーブルまたはディセーブルにするたびに、BGPのベストパス計算がトリガーされます。このオプションを使用すると、グレースフル シャットダウンのウェルノウンコミュニティにおける BGP のベストパス計算の動作を柔軟に制御できます。

始める前に

BGPを有効にしていない場合は、ここで有効にします（**feature bgp**）。

手順の概要

1. **configure terminal**
2. **router bgp** *autonoums-system*
3. （任意） **no graceful-shutdown aware**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch-1(config)# config terminal`
	`switch-1(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router bgp** *autonoums-system*
	例：
	`switch-1(config)# router bgp 100`
	`switch-1(config-router)#`
		ルータ コンフィギュレーション モードを開始し、BGP ルーティング プロセスを設定します。

Step 3	（任意） **no graceful-shutdown aware**
	例：
	`switch-1(config-router)# no graceful-shutdown aware`
	`switch-1(config-router)#`
		このBGPルータでは、GRACEFUL_SHUTDOWN コミュニティを持つすべてのルートに低い優先順位を指定しないという意味です。グレースフル シャットダウン認識機能がディセーブルになっている場合、デフォルト アクションはルートを非優先にします。そのため、コマンドには**no**形式というオプションが存在しており、これを使用すると、グレースフル シャットダウン ルートは非優先になりません。

GRACEFUL_SHUTDOWN コミュニティのピアへの送信の防止

発信ルート更新にルート属性として追加された GRACEFUL_SHUTDOWN コミュニティが不要になった場合は、コミュニティを削除して、指定されたネイバーに送信しなくなります。1 つの使用例は、ルータが自律システム境界にあり、グレースフル シャットダウン機能が自律システム境界の外部に伝播しないようにする場合です。

GRACEFUL_SHUTDOWN がピアに送信されないようにするには、send community オプションを無効にするか、コミュニティを発信ルート マップから削除します。

次の方法の中から 1 つを選択してください。

- 実行コンフィギュレーションで send-community を無効にします。

  例：

  ```
  nxosv2(config-router-neighbor-af)# no send-community standard
  nxosv2(config-router-neighbor-af)#
  ```

  このオプションを使用すると、スイッチは GRACEFUL_SHUTDOWN コミュニティを受信しますが、発信ルートマップを介してダウンストリーム ネイバーに送信されません。すべての標準コミュニティも送信されません。

- 次の手順に従って、発信ルート マップを介して GRACEFUL_SHUTDOWN コミュニティを削除します。

  1. GRACEFUL_SHUTDOWN コミュニティと一致する IP コミュニティ リストを作成します。
  2. GRACEFUL_SHUTDOWN コミュニティと照合する発信ルート マップを作成します。
  3. **set community-list delete** 句を使用して GRACEFUL_SHUTDOWN コミュニティを削除します。

  このオプションを使用すると、コミュニティ リストは GRACEFUL_SHUTDOWN コミュニティと一致し、許可されます。その後、発信ルート マップはコミュニティと照合され、発信ルート マップから削除されます。他のすべてのコミュニティは、問題なく発信ルートマップを通過します。

グレースフル シャットダウン情報の表示

グレースフル シャットダウン機能に関する情報は、次の **show** コマンドで確認できます。

コマンド	アクション
show ip bgp community-list graceful-shutdown	GRACEFUL_SHUTDOWN コミュニティを持つ BGP ルーティング テーブル内のすべてのエントリを表示します。
show running-config bgp	実行中の BGP のデフォルト設定を示します。

コマンド	アクション
show running-config bgp all	グレースフル シャットダウン機能に関する情報など、実行中の BGP 設定のすべての情報を表示します。
show bgp *address-family* **neighbors** *neighbor-address*	機能がピアに設定されている場合、次のように表示されます。 • 指定されたネイバーの graceful-shutdown-activate 機能の状態 • 指定されたネイバーに設定されたグレースフル シャットダウン ルート マップの名前
show bgp process	コンテキストに応じて異なる情報を表示します。 graceful-shutdown-activate オプションがピア コンテキストで設定されている場合、`graceful-shutdown-active` を介して機能の有効または無効状態を示します。 graceful-shutdown-activate オプションがグローバル コンテキストで設定され、graceful-shutdown ルートマップがある場合は、次のように機能の有効状態が表示されます。 • `graceful-shutdown-active` • `graceful-shutdown-aware` • `graceful-shutdown route-map`
show ip bgp *address*	指定されたアドレスについて、次を含む BGP ルーティング テーブル情報を表示します。 • 最適パスとして指定されたアドレスの状態 • 指定されたアドレスが GRACEFUL_SHUTDOWN コミュニティの一部であるかどうか

グレースフル シャットダウンの設定例

次に、グレースフル シャットダウン機能を使用するための設定例を示します。

BGP リンクのグレースフル シャットダウンの設定

次に、ローカルプリファレンスとコミュニティを設定しながらグレースフルシャットダウンを設定する例を示します。

- 指定されたネイバーへのリンクのグレースフル シャットダウン アクティブ化の設定
- ルートへの GRACEFUL_SHUTDOWN コミュニティの追加
- コミュニティとのアウトバウンド ルートに対して set 句のみを使用して gshutPeer という名前のルートマップを設定します。

```
router bgp 100
    neighbor 20.0.0.3 remote-as 200
         graceful-shutdown activate route-map gshutPeer
         address-family ipv4 unicast
              send-community

route-map gshutPeer permit 10
    set local-preference 0
    set community 200:30
```

All-Neighbor BGP リンクのグレースフル シャットダウンの設定

次に例を示します。

- ローカル ルータとそのすべてのネイバーを接続するすべてのリンクに対してグレース フル シャットダウン アクティブ化を設定します。
- GRACEFUL_SHUTDOWN コミュニティをルートに追加しています。
- すべての発信ルートに対して set 句のみを使用して gshutAall という名前のルートマップを設定します。

```
router bgp 200
   graceful-shutdown activate route-map gshutAll

route-map gshutAll permit 10
   set as-path prepend 10 100 110
   set community 100:80

route-map Red permit 10
   set local-pref 20

router bgp 100
   graceful-shutdown activate route-map gshutAll
      router-id 2.2.2.2
         address-family ipv4 unicast
         network 2.2.2.2/32
         neighbor 1.1.1.1 remote-as 100
         update-source loopback0
         address-family ipv4 unicast
            send-community
         neighbor 20.0.0.3 remote-as 200
         address-family ipv4 unicast
            send-community
               route-map Red out
```

この例では、ネイバー 1.1.1.1 に対して gshutAll ルート マップが有効になりますが、ネイバー 20.0.0.3 で設定された発信ルートマップ Red が優先されるため、ネイバー 20.0.0.3 に対しては有効になりません。

ピアテンプレートでのグレースフル シャットダウンの設定

この例では、ピアセッションテンプレートでグレースフルシャットダウン機能を設定します。これはネイバーによって継承されます。

```
router bgp 200
   template peer-session p1
      graceful-shutdown activate route-map gshut_out
   neighbor 1.1.1.1 remote-as 100
      inherit peer-session p1
      address-family ipv4 unicast
         send-community
```

GRACEFUL_SHUTDOWN コミュニティの使用およびインバウンド ルートマップに基づく BGP ルートのフィルタリングとローカル プリファレンスの設定

次に、コミュニティ リストを使用して、GRACEFUL_SHUTDOWN コミュニティを持つ着信ルートをフィルタリングする例を示します。この設定は、Cisco NX-OS 9.3(1) を最小バージョンとして実行していないレガシー スイッチに役立ちます。

次に例を示します。

- GRACEFUL_SHUTDOWN コミュニティを持つルートを許可する IP コミュニティ リスト。
- RM_GSHUT という名前のルート マップは、GSHUT という名前の標準コミュニティ リストに基づいてルートを許可します。
- また、ルート マップは、処理するルートの優先順位を 0 に設定します。これにより、ルータがオフラインになったときに、それらのルートに最適パス計算の優先順位が低くなります。ネイバー (20.0.0.2) からの着信 IPv4 ルートにルート マップが適用されます。

```
ip community-list standard GSHUT permit 65535:0

route-map RM_GSHUT permit 10
   match community GSHUT
   set local-preference 0

router bgp 200
   neighbor 20.0.0.2 remote-as 100
      address-family ipv4 unicast
         send-community
         route-map RM_GSHUT in
```

グレースフル リスタートの設定

グレースフル リスタートを設定し、BGP に対してグレースフル リスタート ヘルパー機能をイネーブルにできます。

（注） Cisco NX-OS リリース 10.1(1) は、より多くの BFD セッションをサポートします。BGP セッションが BFD に関連付けられている場合、ISSU 中にピア接続を維持するために BGP **restart-time** を増やす必要が生じることがあります。

（注） BGP グレースフル リスタートの観点からは、ノードの再起動中にアイドル状態のピアがある場合、最初のベストパスの確立が遅延する可能性があるため、ISSU 中にトラフィック損失が発生する可能性があります。これらのアイドル状態のネイバーをすべて起動するか、それぞれで「shutdown」を構成するか、構成から完全に削除することをお勧めします。

始める前に

BGP をイネーブルにする必要があります（「BGP のイネーブル化」の項を参照）。

VRF を作成します。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. （任意） **timers prefix-peer-timeout** *timeout*
4. **graceful-restart**
5. **graceful-restart** {**restart-time** *time*|**stalepath-time** *time*}
6. **graceful-restart-helper**
7. （任意） **show running-config bgp**
8. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

Step 2	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		自律システム番号を設定して、新しい BGP プロセスを作成します。

	コマンドまたはアクション	目的
Step 3	（任意） **timers prefix-peer-timeout** *timeout* 例： `switch(config-router)# timers prefix-peer-timeout 20`	BGP プレフィックス ピアのタイムアウト値を設定します（秒単位）。デフォルト値は 90 秒です。 （注） このコマンドは、Cisco NX-OS リリース 9.3(3) 以降でサポートされます。
Step 4	**graceful-restart** 例： `switch(config-router)# graceful-restart`	グレースフル リスタートおよびグレースフル リスタート ヘルパー機能をイネーブルにします。このコマンドは、デフォルトでイネーブルになっています。 このコマンドによって、BGP ネイバー セッションの自動通知およびセッション リセットが開始されます。
Step 5	**graceful-restart** {**restart-time** *time*|**stalepath-time** *time*} 例： `switch(config-router)# graceful-restart restart-time 300`	グレースフル リスタート タイマーを設定します。 オプション パラメータは次のとおりです。 • **restart-time**：BGP ピアに送信されたリスタートの最大時間。有効な範囲は 1 ～ 3600 秒です。デフォルトは 120 です。 （注） Cisco NX-OS リリース 10.1(1) は、より多くの BFD セッションをサポートします。BGP セッションが BFD に関連付けられている場合、ISSU 中にピア接続を維持するために BGP **restart-time** を増やす必要が生じることがあります。 • **stalepath-time**：BGP が再起動中の BGP ピアからの古いルートを維持する最大時間有効な範囲は 1 ～ 3600 秒です。デフォルトは 300 です。 NX-OS ソフトウェア リリース 10.2(1) では、BGP セッションがグレースフル リスタート機能をアドバタイズするために、BGP セッションの手動リセットが必要です。NX-OS ソフトウェア リリース 10.2(2) 以降では、このコマンドが有効になっている場合、BGP セッションは、BGP セッションを再起動する必要なく、グレースフル リスタート機能を動的にアドバタイズします。
Step 6	**graceful-restart-helper** 例： `switch(config-router)# graceful-restart restart-time 300`	BGP GR が無効になっている場合、SSO や BGP プロセスの再起動などの特定の GR 対応イベントが N9K でローカルに発生している間、N9K 自体は必ずしも自身の転送状態を保持しません。ただし、GR ヘルパーとして、GR 機能をアドバタイズして再起動して

	コマンドまたはアクション	目的
		いるピアをサポートします。つまり、N9K は、ピアリングがダウンしたことを検出すると（ホールドタイマーの期限切れまたは通知メッセージの受信以外）、ピアを指すルートを失効させ、ピアのEOR（または失効パス タイムアウト）を待機します。ピアが再起動してN9K とのピアリングを再確立すると、ピアは自身のすべてのルートを再アドバタイズし、N9K はBGPおよびルーティング テーブルでそれらのルートを更新します。ピアからEOR を受信するか、または古いパスタイムアウト（どちらか先に発生した方）を受信すると、N9K はそのピアから残りの古いルートをフラッシュします。ヘルパーモードがない場合、N9K は再起動中のリモート ピアから学習したルートを即座にクリアし、トラフィック損失につながる可能性があります。
Step 7	（任意） **show running-config bgp** 例： `switch(config-router)# show running-config bgp`	BGP の設定を表示します。
Step 8	（任意） **copy running-config startup-config** 例： `switch(config-router)# copy running-config startup-config`	この設定変更を保存します。

例

次に、グレースフル リスタートを有効にする例を示します。

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# graceful-restart
switch(config-router)# graceful-restart restart-time 300
switch(config-router)# copy running-config startup-config
```

仮想化の設定

1 つの BGP プロセスを設定し、複数の VRF を作成できます。また、各 VRF で同じ BGP プロセスを使用できます。

始める前に

BGPを有効にする必要があります。

手順の概要

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. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context RemoteOfficeVRF`
	`switch(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。

Step 3	**exit**
	例：
	`switch(config-vrf)# exit`
	`switch(config)#`
		VRF設定モードを終了します。

Step 4	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		自律システム番号を設定して、新しいBGPプロセスを作成します。

Step 5	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf RemoteOfficeVRF`
	`switch(config-router-vrf)#`
		ルータ VRF設定モードを開始し、このBGPインスタンスと VRF を関連付けます。

Step 6	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router-vrf)# neighbor 209.165.201.1 remote-as 65535`
	`switch(config-router--vrf-neighbor)#`
		リモート BGP ピアの IP アドレスおよび AS 番号を設定します。

	コマンドまたはアクション	目的
Step 7	（任意） **copy running-config startup-config** 例： `switch(config-router-vrf-neighbor)# copy running-config startup-config`	この設定変更を保存します。

例

次に、VRF を作成し、VRF でルータ ID を設定する例を示します。

```
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
```

拡張 BGP の設定の確認

BGP の設定を表示するには、次のいずれかの作業を行います。

コマンド	目的
show bgp all [**summary**] [**vrf** *vrf-name*]	すべてのアドレスファミリについて、BGP情報を表示します。
show bgp convergence [vrf *vrf-name]*	すべてのアドレスファミリについて、BGP情報を表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **community** {**regexp** *expression* | [**community**] [**no-advertise**] [**no-export**] [**no-export-subconfed**]} [**vrf** *vrf-name*]	BGP コミュニティと一致する BGP ルートを表示します。
show bgp [**vrf** *vrf-name*] {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **community-list** *list-name* [**vrf** *vrf-name*]	BGP コミュニティ リストと一致する BGP ルートを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **extcommunity** {**regexp** *expression* | **generic** [**non-transitive** | **transitive**] *aa4:nn* [**exact-match**]} [**vrf** *vrf-name*]	BGP拡張コミュニティと一致する BGP ルートを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **extcommunity-list** *list-name* [**exact-match**]} [**vrf** vrf-name]	BGP 拡張コミュニティ リストと一致する BGP ルートを表示します。

コマンド	目的
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **extcommunity-list** *list-name* [**exact-match**]} [**vrf** vrf-name]	BGP ルート ダンプニングの情報を表示します。ルートフラップダンプニング情報を消去するには、**clear bgp dampening** コマンドを使用します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] {**dampening dampened-paths** [**regexp** *expression*]} [**vrf** vrf-name]	BGP ルート ヒストリ パスを表示します。
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **filter-list** *list-name* [**vrf** *vrf-name*]	BGP フィルタ リストの情報を表示します。
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **neighbors** [*ip-address* | *ipv6-prefix*] [**vrf** *vrf-name*]	BGP ピアの情報を表示します。これらのネイバーを消去するには、**clear bgp neighbors** コマンドを使用します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] {**nexthop** | **nexthop-database**} [**vrf** *vrf-name*]	BGP ルート ネクスト ホップの情報を表示します。
show bgp paths	BGP パス情報を表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **policy** *name* [**vrf** *vrf-name*]	BGP ポリシー情報を表示します。ポリシー情報を消去するには、**clear bgp policy** コマンドを使用します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **prefix-list** *list-name* [**vrf** *vrf-name*]	プレフィックスリストと一致する BGP ルートを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **received-paths** [**vrf** *vrf-name*]	ソフト再構成用に保管されている BGP パスを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **regexp** *expression* [**vrf** *vrf-name*]	AS_path 正規表現と一致する BGP ルートを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **route-map** *map-name* [**vrf** *vrf-name*]	ルート マップと一致する BGP ルートを表示します。
show bgp peer-policy *name* [**vrf** *vrf-name*]	BGP ピア ポリシー情報を表示します。
show bgp peer-session *name* [**vrf** *vrf-name*]	BGP ピア セッション情報を表示します。

コマンド	目的
show bgp peer-template *name* [**vrf** *vrf-name*]	BGP ピア テンプレート情報を表示します。ピアテンプレートのすべてのネイバーを消去するには、**clear bgp peer-template** コマンドを使用します。
show bgp process	BGP プロセス情報を表示します。
show bgp {**ipv4** | **ipv6**} **unicast neighbors** *interface*	指定されたインターフェイスの BGP ピアに関する情報を表示します。
show ip bgp neighbors *interface-name*	BGP ピアとして使用されるインターフェイスを表示します。
show ip route *ip-address* **detail vrf all | i bw**	リンク帯域幅の EXTCOMM フィールドを表示します。出力の bw：xx（bw：40 など）は、BGP ピアが帯域幅付きの BGP 拡張属性を送信していることを示します（重み付け ECMP の場合）。
show {**ipv4** | **ipv6**} **bgp** *options*	BGP のステータスと構成情報を表示します。
show {**ipv4** | **ipv6**} **mbgp** *options*	BGP のステータスと構成情報を表示します。
show ipv6 routers interface *interface*	IPv6 ICMP ルータ アドバタイズメントによって学習されたリモート IPv6 ルータのリンク ローカル アドレスを表示します。
show running-configuration bgp	現在実行中の BGP コンフィギュレーションを表示します。

BGP 統計情報のモニタリング

BGP の統計情報を表示するには、次のコマンドを使用します。

コマンド	目的

show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **flap-statistics** [**vrf** *vrf-name*]	BGP ルート フラップの統計情報を表示します。これらの統計情報をクリアするには、**clear bgp flap-statistics** コマンドを使用します。
show bgp {**ipv4** | **ipv6**} **unicast injected-routes**	ルーティング テーブルに挿入されたルートを表示します。
show bgp sessions [**vrf** *vrf-name*]	すべてのピアの BGP セッションを表示します。これらの統計情報をクリアするには、**clear bgp sessions** コマンドを使用します。
show bgp statistics	BGP 統計情報を表示します。

設定例

この例は、個々の BGP ネイバーの BFD をイネーブルにする方法を示します。

```
router bgp 400
  router-id 2.2.2.2
  neighbor 172.16.2.3
    bfd
    remote-as 400
    update-source Vlan1002
    address-family ipv4 unicast
```

この例は、BGP プレフィックス ピアの BFD をイネーブルにする方法を示します。

```
router bgp 400
  router-id 1.1.1.1
  neighbor 172.16.2.0/24
    bfd
    remote-as 400
    update-source Vlan1002
    address-family ipv4 unicast
```

プレフィックス ベース ネイバーの MD5 認証を設定する例を示します。

```
template peer BasePeer-V6
    description BasePeer-V6
    password 3 f4200cfc725bbd28
    transport connection-mode passive
    address-family ipv6 unicast
template peer BasePeer-V4
    bfd
    description BasePeer-V4
    password 3 f4200cfc725bbd28
    address-family ipv4 unicast
--
    neighbor fc00::10:3:11:0/127 remote-as 65006
      inherit peer BasePeer-V6
    neighbor 10.3.11.0/31 remote-as 65006
      inherit peer BasePeer-V4
```

次に、ネイバー ステータスの変化に関するメッセージをグローバルに有効にし、特定のネイバーについてはメッセージを抑制する方法を示します。

```
router bgp 65100
   log-neighbor-changes
     neighbor 209.165.201.1 remote-as 65535
       description test
       address-family ipv4 unicast
         soft-reconfiguration inbound
          disable log-neighbor-changes
```

関連項目

BGP の詳細については、次の項目を参照してください。

- 基本的 BGP の設定 （319 ページ）
- Route Policy Manager の設定 （587 ページ）

その他の参考資料

BGP の実装に関連する詳細情報については、次の項を参照してください。

MIB

MIB	MIB のリンク
BGP に関連する MIB	サポートされている MIB を検索およびダウンロードするには、次の URL にアクセスしてください。 ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

第 12 章

RIP の設定

この章は、次の項で構成されています。

- RIP について （487 ページ）
- RIP の前提条件 （490 ページ）
- RIP に関する注意事項と制約事項 （490 ページ）
- RIP パラメータのデフォルト設定 （491 ページ）
- RIP の設定 （491 ページ）
- RIP の設定の確認 （507 ページ）
- RIP 統計情報の表示 （507 ページ）
- RIP の設定例 （507 ページ）
- 関連項目 （508 ページ）

RIP について

RIP の概要

RIP はユーザ データグラム プロトコル（UDP）データ パケットを使用して、小規模なインターネットワークでルーティング情報を交換します。RIPv2 は IPv4 をサポートします。RIPv2 は RIPv2 プロトコルがサポートするオプションの認証機能を使用します（「RIPv2 の認証」の項を参照）。

RIP では次の 2 種類のメッセージを使用します。

- 要求：他の RIP 対応ルータからのルート アップデートを要求するためにマルチキャスト アドレス 224.0.0.9 に送信されます。
- 応答：デフォルトでは 30 秒間隔で送信されます（「RIP 設定の検証」の項を参照）。ルータも、要求メッセージの受信後に応答メッセージを送信します。応答メッセージには、RIP ルート テーブル全体が含まれます。RIP ルーティング テーブルが 1 つの応答パケットに収まらない場合、RIP は 1 つの要求に対して複数の応答パケットを送信します。

RIP はルーティング メトリックとして、ホップ カウントを使用します。ホップ カウントは、パケットが宛先に到達するまでに、通過できるルータの数です。直接接続されているネットワーク

のメトリックは 1 です。到達不能ネットワークのメトリックは 16 です。RIP はこのようにメトリックの範囲が小さいので、大規模なネットワークに適したルーティング プロトコルではありません。

RIPv2 認証

RIP メッセージに認証を設定して、ネットワークでの不正な、または無効なルーティング更新を防止できます。Cisco NX-OS はシンプルなパスワード、または MD5 認証ダイジェストをサポートしています。

認証キーのキーチェーン管理を使用することによって、インターフェイスごとに RIP 認証を設定できます。キーチェーン管理によって、MD5 認証ダイジェストまたは単純テキストパスワード認証で使用される認証キーの変更を制御できます。キーチェーンの作成の詳細については、『Cisco Nexus 9000 Series NX-OS Security Configuration Guide』を参照してください。

MD5 認証ダイジェストを使用するには、ローカルルータとすべてのリモート RIP ネイバーが共有するパスワードを設定します。Cisco NX-OS は、そのメッセージ自体と暗号化されたパスワードに基づいて MD5 一方向メッセージ ダイジェストを作成し、このダイジェストを RIP メッセージ（要求または応答）とともに送信します。受信側の RIP ネイバーは、同じ暗号パスワードを使用して、ダイジェストを検証します。メッセージが変更されていない場合は、計算が一致し、RIP メッセージは有効と見なされます。

MD5 認証ダイジェストの場合はさらに、ネットワークでメッセージが再送されないように、各 RIP メッセージにシーケンス番号が組み込まれます。

Split Horizon

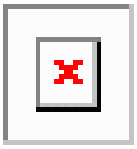
スプリット ホライズンを使用すると、ルートを学習したインターフェイスから RIP がルートをアドバタイズしないようにできます。

スプリット ホライズンは、RIP アップデートおよびクエリー パケットの送信を制御する方法です。インターフェイス上でスプリット ホライズンがイネーブルの場合、Cisco NX-OS はそのインターフェイスから学習した宛先にはアップデート パケットを送信しません。この方法でアップデート パケットを制御すると、ルーティング ループの発生する可能性が小さくなります。

ポイズン リバースを指定してスプリット ホライズンを使用すると、ルートを学習したインターフェイス経由では到達不能であると RIP が学習したルートをアドバタイズするように、インターフェイスを設定できます。

次の図に、ポイズン リバースをイネーブルにしてスプリット ホライズンを指定した、RIP ネットワークの例を示します。

図 *36 :* スプリット ホライズン ポイズン リバースを指定した *RIP*



ルータ C はルート X について学習し、そのルートをルータ B にアドバタイズします。ルータ B はルート X をルータ A にアドバタイズしますが、ルート X の到達不能アップデートをルータ C に送り返します。

デフォルトでは、スプリット ホライズンはすべてのインターフェイスでイネーブルになっています。

ルートのフィルタリング

RIP 対応インターフェイスでルート ポリシーを設定すれば、RIP アップデートをフィルタリングすることができます。Cisco NX-OS は、ルート ポリシーが許可するルートのみでルート テーブルを更新します。

ルート集約

指定したインターフェイスに複数のサマリー集約アドレスを設定できます。ルート集約を使用すると、固有性の強い一連のアドレスをすべての固有アドレスを代表する 1 つのアドレスに置き換えることによって、ルートテーブルを簡素化できます。たとえば、10.1.1.0/24、10.1.2.0/24、および 10.1.3.0/24 というアドレスを 1 つの集約アドレス 10.1.0.0/16 に置き換えることができます。

RIP はルーティング テーブルに含まれている固有性の強いルートが多いほど、固有性の強いルートの最大メトリックと同じメトリックのインターフェイスからのサマリー アドレスをアドバタイズします。

（注） Cisco NX-OS は、自動ルート集約をサポートしていません。

ルートの再配布

RIP を使用すると、スタティック ルートや他のプロトコルからのルートを再配布できます。再配布を指定したルート マップを設定して、どのルートが RIP に渡されるかを制御する必要があります。ルート ポリシーを使用すると、宛先、送信元プロトコル、ルート タイプ、ルート タグなどの属性に基づいて、ルートをフィルタリングできます。詳細については、Route Policy Manager の設定 （587 ページ）を参照してください。

RIP ルーティング ドメインにルートを再配布しても、デフォルトでは Cisco NX-OS がそのつど、RIP ルーティング ドメインにデフォルト ルートを再配布することはありません。RIP にデフォルト ルートを生成し、ルート ポリシーでそのルートを制御できます。

RIP にインポートされたすべてのルートに使用する、デフォルトのメトリックも設定できます。

ロード バランシング

ロード バランシングを使用すると、ルータは、宛先アドレスから等距離内にあるすべてのルータのネットワーク ポートにトラフィックを分散できます。ロード バランシングは、ネットワークセグメントの使用率を向上させ、有効ネットワーク帯域幅を増加させます。

Cisco NX-OS は、等コスト マルチパス（ECMP）機能をサポートします。RIP ルート テーブルおよびユニキャスト RIB の等コスト パスは最大 16 です。これらのパスの一部または全部でトラフィックのロード バランシングが行われるように、RIP を設定できます。

RIP のハイ アベイラビリティ

Cisco NX-OS は、RIP のステートレス リスタートをサポートします。リブートまたはスーパーバイザ スイッチオーバー後に、Cisco NX-OS が実行コンフィギュレーションを適用し、RIP がただちに要求パケットを送信して、ルーティング テーブルに再入力します。

RIP 仮想化のサポート

Cisco NX-OS は、同一システム上で動作する複数の RIP プロトコル インスタンスをサポートします。RIP は、仮想ルーティングおよび転送（VRF）インスタンスをサポートします。

RIP の前提条件

RIP を使用するには、次の前提条件を満たしている必要があります。

- RIP をイネーブルにします（「RIP のネーブル化」セクションを参照）。

RIP に関する注意事項と制約事項

RIP には、次の注意事項および制限事項があります。

- プレフィックス リスト内の名前は、大文字と小文字が区別されません。一意の名前を使用することを推奨します。大文字と小文字を変更しただけの名前は使用しないでください。たとえば、CTCPrimaryNetworks と CtcPrimaryNetworks は 2 つの異なるエントリではありません。
- Cisco NX-OS は、RIPv1 をサポートしません。Cisco NX-OS がRIPv1 パケットを受信した場合、メッセージを記録してパケットをドロップします。
- Cisco NX-OS は、RIPv1 ルータとの隣接関係を確立しません。
- RIP はトンネル インターフェイスではサポートされていません。

（注） RIP は、255 以下の 8 ビット KeyID のみをサポートします。これは、RIP で認証を設定するときに使用される keyID です。

RIP パラメータのデフォルト設定

次の表に、RIP パラメータのデフォルト設定値を示します。

デフォルトの RIP パラメータ

パラメータ	デフォルト
ロード バランシングを行う最大パス数	16
RIP 機能	ディセーブル
スプリット ホライズン	有効（Enabled）

RIP の設定

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

RIP のイネーブル化

RIP を設定するには、その前に RIP を有効にする必要があります。

手順の概要

1. **configure terminal**
2. **[no] feature rip**
3. （任意） **show feature**
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	[**no**] **feature rip** 例： `switch(config)# feature rip`	RIP 機能を有効にします。
Step 3	（任意） **show feature** 例： `switch(config)# show feature`	有効および無効にされた機能を表示します。
Step 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

RIP インスタンスの作成

RIP インスタンスを作成し、そのインスタンスのアドレス ファミリを設定できます。

始める前に

RIP をイネーブルにします（「RIP のネーブル化」セクションを参照）。

手順の概要

1. **configure terminal**
2. **[no] router rip** *instance-tag*
3. **address-family ipv4 unicast**
4. （任意） **show ip rip** [**instance** *instance-tag*] [**vrf** *vrf-name*]
5. （任意） **distance** *value*
6. （任意） **maximum-paths** *number*
7. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**[no] router rip** *instance-tag*
	例：
	`switch(config)# router RIP Enterprise`
	`switch(config-router)#`
		instance tag を設定して、新しい RIP インスタンスを作成します。

Step 3	**address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		この RIP インスタンスのアドレス ファミリを設定し、アドレス ファミリ コンフィギュレーション モードを開始します。

Step 4	（任意） **show ip rip** [**instance** *instance-tag*] [**vrf** *vrf-name*] 例： `switch(config-router-af)# show ip rip`	すべての RIP インスタンスの RIP 要約情報を表示します。
Step 5	（任意） **distance** *value* 例： `switch(config-router-af)# distance 30`	RIP のアドミニストレーティブ ディスタンスを設定します。範囲は 1 ～ 255 です。デフォルトは 120 です。「アドミニストレーティブ ディスタンス」のセクションを参照してください。
Step 6	（任意） **maximum-paths** *number* 例： `switch(config-router-af)# maximum-paths 6`	RIP がルート テーブルで維持する等コスト パスの最大数を設定します。有効な範囲は 1 ～ 64 です。デフォルトは 16 です。
Step 7	（任意） **copy running-config startup-config** 例： `switch(config-router-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、IPv4 に対応する RIP インスタンスを作成し、ロード バランシングのための等コスト パス数を設定する例を示します。

```
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
```

RIP インスタンスの再起動

RIPインスタンスを再起動し、インスタンスに関連付けられているすべてのネイバーを削除できます。

RIP インスタンスを再起動し、関連付けられたすべてのネイバーを削除するには、グローバル設定モードで次のコマンドを使用します。

手順の概要

1. **restart rip** *instance-tag*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**restart rip** *instance-tag* 例： `switch(config)# restart rip Enterprise`	RIP インスタンスを再起動し、すべてのネイバーを削除します。

インターフェイスでの RIP の設定

始める前に

RIP をイネーブルにします（「RIP のネーブル化」セクションを参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ip router rip** *instance-tag*
4. （任意） **show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot/port*] [**vrf** *vrf-name*] [**detail**]
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**ip router rip** *instance-tag* 例： `switch(config-if)# ip router rip Enterprise`	このインターフェイスを RIP インスタンスに関連付けます。
Step 4	（任意） **show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot/port*] [**vrf** *vrf-name*] [**detail**] 例： `switch(config-if)# show ip rip Enterprise tethernet 1/2`	インターフェイスの RIP 情報を表示します。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

例

次に、RIP インスタンスに Ethernet 1/2 インターフェイスを追加する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ip router rip Enterprise
switch(config)# copy running-config startup-config
```

RIP 認証の設定

インターフェイスに RIP パケットの認証を設定できます。

始める前に

RIP をイネーブルにします（「RIP のネーブル化」セクションを参照）。

認証をイネーブルにする前に、必要に応じてキーチェーンを設定します。キー チェーンの実装の詳細については、『Cisco Nexus 9000 Series NX-OS Security Configuration Guide』を参照してください。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ip rip authentication mode** {**text** | **md5**}
4. **ip rip authentication key-chain** *key*
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**ip rip authentication mode** {**text** | **md5**} 例： `switch(config-if)# ip rip authentication mode md5`	クリアテキストまたは MD5 認証ダイジェストとして、このインターフェイスにおける RIP 認証タイプを設定します。
Step 4	**ip rip authentication key-chain** *key* 例： `switch(config-if)# ip rip authentication key-chain RIPKey`	このインターフェイス上で RIP に使用する認証キーを設定します。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

例

次に、キーチェーンを作成し、RIP インターフェイス上で MD5 認証を設定する例を示します。

```
switch# configure terminal
switch(config)# key chain RIPKey
switch(config-keychain)# key 2
switch(config-keychain-key)# accept-lifetime 00:00:00 Jan 01 2000 infinite
switch(config-keychain-key)# send-lifetime 00:00:00 Jan 01 2000 infinite
switch(config-keychain-key)# exit
switch(config-keychain)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# ip rip authentication mode md5
switch(config-if)# ip rip authentication key-chain RIPKey
switch(config-if)# copy running-config startup-config
```

パッシブ インターフェイスの設定

インターフェイスを受動モードに設定することによって、ルートを受信するが、ルート アップデートの送信は行わないように RIP インターフェイスを設定できます。

受動モードで RIP インターフェイスを設定するには、インターフェイス設定モードで次のコマンドを使用します。

手順の概要

1. **ip rip passive-interface**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**ip rip passive-interface** 例: `switch(config-if)# ip rip passive-interface`	インターフェイスを受動モードに設定します。

ポイズン リバースを指定したスプリット ホライズンの設定

インターフェイスの設定でポイズン リバースをイネーブルにすると、RIP が学習したルートについて、ルートを学習したインターフェイス経由では到達不能であることをアドバタイズできます。

インターフェイス上で、ポイズン リバースを指定してスプリット ホライズンを設定するには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **ip rip poison-reverse**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**ip rip poison-reverse** 例： `switch(config-if)# ip rip poison-reverse`	ポイズン リバースを指定してスプリット ホライズンをイネーブルにします。ポイズン リバースを指定したスプリット ホライズンは、デフォルトでディセーブルです。

ルート集約の設定

ルーティングテーブルでサマリーアドレスによって表される集約アドレスを作成できます。Cisco NX-OS は、固有性の強いすべてのルートの中でメトリックが最小のサマリー アドレス メトリックをアドバタイズします。

インターフェイス上でサマリー アドレスを設定するには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **ip rip summary-address** *ip-prefix/mask-len*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**ip rip summary-address** *ip-prefix/mask-len* 例： `switch(config-if)# ip rip summary-address 1.1.1.1/32`	IPv4 アドレスに対応する、RIP 用のサマリー アドレスを設定します。

ルートの再配布の設定

別のルーティング プロトコルからのルーティング情報を受け入れて、RIP ネットワークを通じてその情報を再配布するように、RIP を設定できます。再配布されたルートを任意で、デフォルトルートとして割り当てることができます。

始める前に

RIP を有効にします（「RIP の有効化」セクションを参照）。

再配布を設定する前に、ルート マップを設定します。ルート マップの設定の詳細については、「ルート マップの設定 」セクションを参照してください。

手順の概要

1. **configure terminal**
2. **router rip** *instance-tag*
3. **address-family ipv4 unicast**
4. **redistribute** {**bgp** *as* | **direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} **route-map** *map-name*
5. （任意） **default-information originate** [**always**] [**route-map** *map-name*]
6. （任意） **default-metric** *value*
7. （任意） **show ip rip route** [*ip-prefix* [**longer-prefixes** | **shorter-prefixes**]] [**vrf** *vrf-name*] [**summary**]
8. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router rip** *instance-tag*
	例：
	`switch(config)# router rip Enterprise`
	`switch(config-router)#`
		instance tag を設定して、新しい RIP インスタンスを作成します。

Step 3	**address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		アドレスファミリ コンフィギュレーション モードに入ります。

Step 4	**redistribute** {**bgp** *as* | **direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} **route-map** *map-name* 例： `switch(config-router-af)# redistribute eigrp 201 route-map RIPmap`	他のプロトコルからのルートを RIP に再配布します。

	コマンドまたはアクション	目的
Step 5	（任意） **default-information originate** [**always**] [**route-map** *map-name*] 例： `switch(config-router-af)# default-information originate always`	RIPにデフォルトルートを生成し、必要に応じてルート マップにより制御します。
Step 6	（任意） **default-metric** *value* 例： `switch(config-router-af)# default-metric 2`	再配布されたすべてのルートにデフォルト メトリックを設定します。有効な範囲は 1 ～ 15 です。デフォルトは 1 です。
Step 7	（任意） **show ip rip route** [*ip-prefix* [**longer-prefixes** | **shorter-prefixes**]] [**vrf** *vrf-name*] [**summary**] 例： `switch(config-router-af)# show ip rip route`	RIP のルートを表示します。
Step 8	（任意） **copy running-config startup-config** 例： `switch(config-router-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、EIGRP を RIP に再配布する例を示します。

```
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
```

Cisco IOS RIP との互換性のため、Cisco NX-OS RIP を設定

Cisco NX-OS RIP を、ルートがアドバタイズされ、処理される方法で Cisco IOS RIP のように動作するよう設定できます。

直接接続されたルートが、Cisco NX-OS RIP ではコスト 1 として処理され、Cisco IOS RIP ではコスト 0 として処理されます。ルートが Cisco NX-OS RIP でアドバタイズされる場合、受信デバイスはすべての受信ルートに +1 の最小のコストを増加し、ルーティング テーブルにルートをインストールします。Cisco IOS RIP において、このコストの増加は送信側ルータで実行され、受信側ルータは変更なしでルートをインストールします。Cisco NX-OS および Cisco IOS デバイスの両方が連携しているときに、この動作の違いにより問題が発生する可能性があります。Cisco IOS RIP など、ルートをアドバタイズし、処理するために、Cisco NX-OS RIP の設定に応じて、次の互換性の問題を回避できます。

始める前に

RIP をイネーブルにします（「RIP のネーブル化」セクションを参照）。

手順の概要

1. **configure terminal**
2. **router rip** *instance-tag*
3. [**no**] **metric direct 0**
4. **address-family ipv4 unicast**
5. **vrf abc**
6. **flush-routes**
7. **isolate**
8. **shutdown**
9. （任意） **show running-config rip**
10. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router rip** *instance-tag*
	例：
	`switch(config)# router rip 100`
	`switch(config-router)#`
		instance tag を設定して、新しい RIP インスタンスを作成します。インスタンス タグには 100、201、または 20 文字までの英数字を入力できます。

Step 3	[**no**] **metric direct 0** 例： `switch(config-router)# metric direct 0`	ルートがアドバタイズされ、処理される方法でCisco IOS RIP と Cisco NX-OS RIP が互換性を持つようにするため、直接接続するルータすべてをデフォルトであるコスト 1 の代わりにコスト 0 で設定します。 （注） このコマンドは、Cisco IOS デバイスを含む RIP ネットワークに存在するすべての Cisco NX-OS デバイスで設定する必要があります。
Step 4	**address-family ipv4 unicast** 例： `switch(config-router)# address-family ipv4 unicast`	この RIP インスタンスの VRF アドレス ファミリを設定します。

	コマンドまたはアクション	目的
Step 5	**vrf abc** 例： `switch(config-router)# vrf abc`	新しい VRF を作成します。
Step 6	**flush-routes** 例： `switch(config-router)# flush-routes`	ルートを削除します。
Step 7	**isolate** 例： `switch(config-router)# isolate`	RIP の観点からルータを分離します
Step 8	**shutdown** 例： `switch(config-router)# shutdown`	RIP インスタンスをシャットダウンします。 （注） show run の出力は、構成された shutdown コマンドがプロセス中であり、vrf も作成されている場合とは異なります。シャットダウン構成は、VRF 構成の上に表示されます。

Step 9	（任意） **show running-config rip**
	例：
	`switch(config-router)# show running-config rip`
	`!Command: show running-config rip`
	`!Running configuration last done at: Tue Dec 17 01:13:28 2024`
	`!Time: Tue Dec 17 01:13:31 2024`
		現在実行中の RIP コンフィギュレーションを表示します。

Step 10	（任意） **copy running-config startup-config** 例： `switch(config-router)# copy running-config startup-config`	この設定変更を保存します。

例

次に、すべての直接ルートをコスト 0 からコスト 1 に返すことによって、Cisco IOS RIP と Cisco NX-OS RIP の互換性をディセーブルにする例を示します。

```
switch# configure terminal
switch(config)# router rip 100
switch(config-router)# no metric direct 0
switch(config-router)# show running-config rip
switch(config-router)# copy running-config startup-config
```

仮想化の設定

複数の RIP インスタンスを設定し、複数の VRF を作成し、同じまたは複数の RIP インスタンスを各 VRF で使用するようにできます。VRF に RIP インターフェイスを割り当てます。

（注） インターフェイスの VRF を設定した後に、インターフェイスの他のすべてのパラメータを設定します。インターフェイスの VRF を設定すると、そのインターフェイスのすべての設定が削除されます。

始める前に

RIP をイネーブルにします（「RIP のネーブル化」の項を参照）。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **exit**
4. **router rip** *instance-tag*
5. **vrf** *vrf-name*
6. （任意） **address-family ipv4 unicast**
7. （任意） **redistribute** {**bgp** *as* | **direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} **route-map** *map-name*
8. **interface ethernet** *slot*/*port*
9. **vrf member** *vrf-name*
10. **ip address** *ip-prefix*/*length*
11. **ip router rip** *instance-tag*
12. （任意） **show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot*/*port*] [**vrf** *vrf-name*]
13. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**vrf context** *vrf-name* 例：	新しい VRF を作成し、VRF 設定モードを開始します。

	コマンドまたはアクション	目的
	`switch(config)# vrf context RemoteOfficeVRF` `switch(config-vrf)#`	

Step 3	**exit**
	例：
	`switch(config-vrf)# exit`
	`switch(config)#`
		VRF設定モードを終了します。

Step 4	**router rip** *instance-tag*
	例：
	`switch(config)# router rip Enterprise`
	`switch(config-router)#`
		instance tag を設定して、新しい RIP インスタンスを作成します。

Step 5	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf RemoteOfficeVRF`
	`switch(config-router-vrf)#`
		新しい VRF を作成します。

Step 6	（任意） **address-family ipv4 unicast**
	例：
	`switch(config-router-vrf)# address-family ipv4 unicast`
	`switch(config-router-vrf-af)#`
		この RIP インスタンスの VRF アドレス ファミリを設定します。

Step 7	（任意） **redistribute** {**bgp** *as* | **direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} **route-map** *map-name* 例： `switch(config-router-vrf-af)# redistribute eigrp 201 route-map RIPmap`	他のプロトコルからのルートを RIP に再配布します。 ルートマップの詳細については、ルートマップの設定 （612 ページ）を参照してください。

Step 8	**interface ethernet** *slot/port*
	例：
	`switch(config-router-vrf-af)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 9	**vrf member** *vrf-name* 例： `switch(config-if)# vrf member RemoteOfficeVRF`	このインターフェイスを VRF に追加します。
Step 10	**ip address** *ip-prefix/length* 例： `switch(config-if)# ip address 192.0.2.1/16`	このインターフェイスの IP アドレスを設定します。このステップは、このインターフェイスを VRF に割り当てたあとに行う必要があります。
Step 11	**ip router rip** *instance-tag* 例：	このインターフェイスを RIP インスタンスに関連付けます。

	コマンドまたはアクション	目的
	`switch(config-if)# ip router rip Enterprise`	
Step 12	（任意） **show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot*/*port*] [**vrf** *vrf-name*] 例： `switch(config-if)# show ip rip Enterprise ethernet 1/2`	VRF のインターフェイスに関する RIP 情報を表示します。
Step 13	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

例

次に、VRF を作成して、その VRF にインターフェイスを追加する例を示します。

```
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)# 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
```

RIP の調整

ネットワーク要件に適合するように RIP を調整できます。RIP では複数のタイマーを使用して、ルーティング アップデート間隔、ルートが無効になるまでの時間の長さ、およびその他のパラメータを決定します。これらのタイマーを調整すると、インターネットワークのニーズに適合するように、ルーティング プロトコルのパフォーマンスを調整できます。

（注） ネットワーク上のすべての RIP 対応ルータで、RIP タイマーに同じ値を設定する必要があります。

RIP を調整するには、アドレス ファミリ コンフィギュレーション モードで次のオプション コマンドを使用します。

コマンド	目的
timers basic *update timeout holddown garbage-collection* 例： `switch(config-router-af)# timers basic 40 120 120 100`	RIP タイマーを秒数で設定します。パラメータは次のとおりです。 • *update*：指定できる範囲は 5 ～任意の正の整数。デフォルトは 30 です。 • *timeout*：ルートの無効を宣言するまでに、Cisco NX-OS が待機する時間。タイムアウト インターバルが終了するまでに、このルートのアップデート情報を Cisco NX-OS が受信しなかった場合、Cisco NX-OS はルートの無効を宣言します。指定できる範囲は 1 ～任意の正の整数です。デフォルトは 180 です。 • *holddown*：無効ルートに関するよりよいルート情報を Cisco NX-OS が無視する時間。指定できる範囲は 0 ～任意の正の整数です。デフォルトは 180 です。 • *garbage-collection*：Cisco NX-OS がルートを無効として表示してから、Cisco NX-OS がそのルートをルーティング テーブルから削除するまでの時間。指定できる範囲は 1 ～任意の正の整数です。デフォルトは 120 です。

RIP を調整するには、インターフェイス コンフィギュレーション モードで次のオプション コマンドを使用します。

コマンド	目的
ip rip metric-offset *value* 例： `switch(config-if)# ip rip metric-offset 10`	このインターフェイスで受信する各ルートのメトリックに値を追加します。有効な範囲は 1 ～ 15 です。デフォルトは 1 です。
ip rip route-filter {**prefix-list** *list-name* | **route-map** *map-name* | [**in** | **out**]	着信または発信 RIP アップデートをフィルタリングするための、ルートマップを指定します。
例： `switch(config-if)# ip rip route-filter route-map InputMap in`	

RIP の設定の確認

RIP の設定を表示するには、次のいずれかの作業を行います。

コマンド	目的
show ip rip instance [*instance-tag*] [**vrf** *vrf-name*]	RIP インスタンスの状態を表示します。
show ip rip [**instance** *instance-tag*] **interface** *slot/port* **detail** [**vrf** *vrf-name*]	インターフェイスの RIP ステータスを表示します。
show ip rip [**instance** *instance-tag*] **neighbor** [*interface-type number*] [**vrf** *vrf-name*]	RIP ネイバー テーブルを表示します。
show ip rip [**instance** *instance-tag*] **route** [*ip-prefix/length* [**longer-prefixes** | **shorter-prefixes**]] [**summary**] [**vrf** *vrf-name*]	RIP ルート テーブルを表示します。
show running-configuration rip	現在実行中の RIP コンフィギュレーションを表示します。

RIP 統計情報の表示

RIP の統計情報を表示するには、次のコマンドを使用します。

コマンド	目的
show ip rip [**instance** *instance-tag*] **policy statistics redistribute** {**bgp** *as* | **direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} [**vrf** *vrf-name*]	RIP ポリシー統計情報を表示します。
show ip rip [**instance** *instance-tag*] **statistics** *interface-type number* [**vrf** *vrf-name*]	RIP の統計情報を表示します。

clear rip policy statistics redistribute *protocol process-tag* コマンドを使用して、ポリシー統計情報をクリアします。

clear ip rip statistics コマンドを使用し、 して、RIP 統計情報をクリアします。

RIP の設定例

VRF で Enterprise RIP インスタンスを作成し、その RIP インスタンスにイーサネット インターフェイス 1/2 の例を示します。さらに、enthernet interface 1/2 の認証を設定し、この RIP ドメインに EIGRP を再配布する例も示します

```
vrf context NewVRF
!
 feature rip
```

```
 router rip Enterprise
  vrf NewVRF
   address-family ipv4 unicast
    redistribute eigrp 201 route-map RIPmap
    maximum-paths 10
!
interface ethernet 1/2
 vrf member NewVRF
 ip address 192.0.2.1/16
 ip router rip Enterprise
 ip rip authentication mode md5
 ip rip authentication key-chain RIPKey
```

次の例は、有効な keyID 設定を示しています。

```
### Valid
key-chain kc1
key 255
key-string ...
```

関連項目

ルートマップの詳細については、Route Policy Manager の設定（587 ページ）を参照してください。

第 13 章

RIPng の設定

この章は、次の項で構成されています。

- RIPng について （509 ページ）
- RIPng の前提条件 （511 ページ）
- RIPng のガイドラインと制限事項 （512 ページ）
- RIPng パラメータのデフォルト設定 （512 ページ）
- RIPng の設定 （512 ページ）
- RIPng 構成の確認 （522 ページ）
- RIPng 統計の表示 （523 ページ）
- RIPng の設定例 （523 ページ）
- 関連項目 （523 ページ）

RIPng について

RIPng の概要

RIPng はユーザ データグラム プロトコル（UDP）データ パケットを使用して、小規模なインターネットワークでルーティング情報を交換します。

RIPng は IPv6 をサポートし、次の 2 つのメッセージ タイプを使用します。

- 要求：他の RIPng 対応ルータからのルート アップデートを要求するためにマルチキャスト アドレス FF02::9 に送信されます。
- 応答：デフォルトでは 30 秒間隔で送信されます（RIPng 構成の確認 （522 ページ）セクションを参照）。ルータも、要求メッセージの受信後に応答メッセージを送信します。応答メッセージには、RIPng ルート テーブル全体が含まれます。RIPng ルーティング テーブルが 1 つの応答パケットに収まらない場合、RIPng は 1 つの要求に対して複数の応答パケットを送信します。

RIPng はルーティング メトリックとして、ホップ カウントを使用します。ホップ カウントは、パケットが宛先に到達するまでに、通過できるルータの数です。直接接続されているネットワーク

のメトリックは 1 です。到達不能ネットワークのメトリックは 16 です。RIPng はこのようにメトリックの範囲が小さいので、大規模なネットワークに適したルーティング プロトコルではありません。

Split Horizon

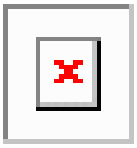
スプリット ホライズンを使用すると、ルートを学習したインターフェイスから RIPng がルートをアドバタイズしないようにできます。

スプリット ホライズンは、RIPng アップデートおよびクエリー パケットの送信を制御する方法です。インターフェイス上でスプリット ホライズンがイネーブルの場合、Cisco NX-OS はそのインターフェイスから学習した宛先にはアップデート パケットを送信しません。この方法でアップデート パケットを制御すると、ルーティング ループの発生する可能性が小さくなります。

ポイズン リバースを指定してスプリット ホライズンを使用すると、RIPng が学習したルートについて、ルートを学習したインターフェイス経由では到達不能であるとアドバタイズするように、インターフェイスを設定できます。

次の図に、ポイズン リバースを有効にしてスプリット ホライズンを指定した、RIPng ネットワークの例を示します。

図 *37*：スプリット ホライズン ポイズン リバースを指定した *RIPng*



ルータ C はルート X について学習し、そのルートをルータ B にアドバタイズします。ルータ B はルート X をルータ A にアドバタイズしますが、ルート X の到達不能アップデートをルータ C に送り返します。

デフォルトでは、スプリット ホライズンはすべてのインターフェイスでイネーブルになっています。

ルートのフィルタリング

RIPng 対応インターフェイスでルート ポリシーを構成すれば、RIPng アップデートをフィルタリングすることができます。Cisco NX-OS は、ルート ポリシーが許可するルートのみでルート テーブルを更新します。

ロード バランシング

ロード バランシングを使用すると、ルータは、宛先アドレスから等距離内にあるすべてのルータのネットワーク ポートにトラフィックを分散できます。ロード バランシングは、ネットワーク セグメントの使用率を向上させ、有効ネットワーク帯域幅を増加させます。

Cisco NX-OS は、等コスト マルチパス（ECMP）機能をサポートします。RIP ルート テーブルおよびユニキャスト RIPng の等コスト パスは最大 16 です。これらのパスの一部または全部でトラフィックのロード バランシングが行われるように、RIPng を設定できます。

デフォルトの情報の発信元と生成

Cisco NX-OS は、RIPng IPv6 のデフォルト情報の発信と生成をサポートしています。

デフォルト ルートを Routing Information Protocol（RIP）に生成するには、ルータ アドレスファミリ 構成 モードで default-information originate コマンドを使用します。この機能をディセーブルにするには、このコマンドの **no** 形式を使用します。

default-information originate [**always**] [**route-map** *map-name*]

no default-information originate

（注） ルートが RIP ルーティング情報ベース、つまり RIP 内部 RIB に存在しない場合は、always キーワードを使用してデフォルト ルートを生成します。route-map キーワードを map-name 変数とともに使用して、ルートがルートマップによって許可されている場合にのみデフォルトルートを生成します。マップ名は、63 文字以下の任意の英数字文字列です。originate を使用して、定期的な更新とともにデフォルト ルートを送信します。

次に、条件ルートマップをパスしたすべてのルートに対して、デフォルトルートを生成する例を示します。

switch(config)# **router rip Enterprise**

switch(config-router)# **address-family ipv6 unicast**

switch(config-router-af)# **default-information originate route-map Condition**

RIPng の高可用性

Cisco NX-OS は、RIPng のステートレス リスタートをサポートします。リブートまたはスーパーバイザ スイッチオーバー後に、Cisco NX-OS が実行構成を適用し、RIPng がただちに要求パケットを送信して、ルーティング テーブルに再入力します。

RIPng の仮想化のサポート

Cisco NX-OS は、同一システム上で動作する複数の RIPng プロトコル インスタンスをサポートします。RIPng は、仮想ルーティングおよび転送（VRF）インスタンスをサポートします。

RIPng の前提条件

RIPng には、次の前提条件があります。

- RIPng を有効にする必要があります（RIPng の有効化 （512 ページ） セクションを参照）。

RIPng のガイドラインと制限事項

RIPng には、次の構成時のガイドラインと制限事項があります。

- Cisco NX-OS リリース 10.2(3)F 以降、Cisco Nexus 9300 および 9500 シリーズ プラットフォーム スイッチで、IPv6 をサポートするために RIPng 機能が導入されました。
- プレフィックス リスト内の名前は、大文字と小文字が区別されません。一意の名前を使用することを推奨します。大文字と小文字を変更しただけの名前は使用しないでください。たとえば、CTCPrimaryNetworks と CtcPrimaryNetworks は 2 つの異なるエントリではありません。
- Cisco NX-OS は、RIPv1 をサポートしません。Cisco NX-OS がRIPv1 パケットを受信した場合、メッセージを記録してパケットをドロップします。
- Cisco NX-OS は、RIPv1 ルータとの隣接関係を確立しません。

RIPng パラメータのデフォルト設定

次の表に、RIPng パラメータのデフォルト設定値を示します。

デフォルトの RIPng パラメータ

パラメータ	デフォルト
ロード バランシングを行う最大パス数	16
RIPng 機能	ディセーブル
スプリット ホライズン	有効（Enabled）

RIPng の設定

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

RIPng の有効化

RIPng を構成する前に、RIPng を有効にする必要があります。

手順の概要

1. **configure terminal**
2. [**no**] **feature rip**
3. （任意） **show feature**
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	[**no**] **feature rip** 例： `switch(config)# feature rip`	RIPng 機能を有効にします。
Step 3	（任意） **show feature** 例： `switch(config)# show feature`	有効および無効にされた機能を表示します。
Step 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

RIPng インスタンスの作成

RIPngインスタンスを作成し、そのインスタンスのアドレスファミリを構成することができます。

始める前に

RIPng を有効にする必要があります（RIPng の有効化 （512 ページ） セクションを参照）。

手順の概要

1. **configure terminal**
2. **[no] router rip** *instance-tag*
3. **address-family ipv6 unicast**
4. （任意） **show ipv6 rip** [**instance** *instance-tag*] [**vrf** *vrf-name*]
5. （任意） **distance** *value*

6. （任意） **maximum-paths** *number*
7. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**[no] router rip** *instance-tag*
	例：
	`switch(config)# router RIP Enterprise`
	`switch(config-router)#`
		instance-tag を構成して、新しい RIPng インスタンスを作成します。

Step 3	**address-family ipv6 unicast**
	例：
	`switch(config-router)# address-family ipv6 unicast`
	`switch(config-router-af)#`
		この RIPng インスタンスのアドレス ファミリを構成し、アドレスファミリ構成モードを開始します。

Step 4	（任意） **show ipv6 rip** [**instance** *instance-tag*] [**vrf** *vrf-name*] 例： `switch(config-router-af)# show ipv6 rip`	すべての RIPng インスタンスの RIPng 情報の概要を表示します。
Step 5	（任意） **distance** *value* 例： `switch(config-router-af)# distance 30`	RIPng のアドミニストレーティブ ディスタンスを設定します。範囲は 1 ～ 255 です。デフォルトは 120 です。「アドミニストレーティブ ディスタンス」のセクションを参照してください。
Step 6	（任意） **maximum-paths** *number* 例： `switch(config-router-af)# maximum-paths 6`	RIPng がルート テーブルで維持する等コスト パスの最大数を構成します。有効な範囲は 1 ～ 64 です。デフォルトは 16 です。
Step 7	（任意） **copy running-config startup-config** 例： `switch(config-router-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、IPv6 に対応する RIPng インスタンスを作成し、ロード バランシングのための等コスト パス数を設定する例を示します：

```
switch# configure terminal
switch(config)# router rip Enterprise
switch(config-router)# address-family ipv6 unicast
switch(config-router-af)# max-paths 10
switch(config-router-af)# copy running-config startup-config
```

RIPng インスタンスの再起動

RIPng インスタンスを再起動すれば、インスタンスに関連付けられているすべてのネイバーを削除できます。

RIPng インスタンスを再起動し、関連付けられたすべてのネイバーを削除するには、グローバル構成モードで次のコマンドを使用します。

手順の概要

1. **restart rip** *instance-tag*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**restart rip** *instance-tag* 例： `switch(config)# restart rip Enterprise`	RIPng インスタンスを再起動し、すべてのネイバーを削除します。

インターフェイス上での RIPng の構成

始める前に

RIPng を有効にする必要があります（RIPng の有効化 （512 ページ） セクションを参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ipv6 router rip** *instance-tag*
4. （任意） **show ipv6 rip** [**instance** *instance-tag*] **interface** [*interface-type slot/port*] [**vrf** *vrf-name*] [**detail**]

5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**ipv6 router rip** *instance-tag* 例： `switch(config-if)# ipv6 router rip Enterprise`	このインターフェイスを RIPng インスタンスと関連付けます。
Step 4	（任意） **show ipv6 rip** [**instance** *instance-tag*] **interface** [*interface-type slot/port*] [**vrf** *vrf-name*] [**detail**] 例： `switch(config-if)# show ipv6 rip Enterprise ethernet 1/2`	インターフェイスの RIPng 情報を表示します。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

例

次に、RIPng インスタンスに Ethernet 1 / 2 インターフェイスを追加する例を示します：

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ipv6 router rip Enterprise
switch(config)# copy running-config startup-config
```

ポイズン リバースを指定したスプリット ホライズンの設定

インターフェイスの設定でポイズン リバースを有効にすると、RIP が学習したルートについて、ルートを学習したインターフェイス経由では到達不能であるとをアドバタイズできます。

インターフェイス上で、ポイズン リバースを指定してスプリット ホライズンを設定するには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **ipv6 rip poison-reverse**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**ipv6 rip poison-reverse** 例： `switch(config-if)# ipv6 rip poison-reverse`	ポイズン リバースを指定してスプリット ホライズンをイネーブルにします。ポイズン リバースを指定したスプリット ホライズンは、デフォルトでディセーブルです。

Cisco IOS RIPng との互換性のための Cisco NX-OS RIPng の構成

Cisco NX-OS RIPng は、ルートのアドバタイズと処理において、Cisco IOS RIPng のように動作するよう構成できます。

直接接続されたルートは、Cisco NX-OS RIPng ではコスト 1 として処理され、Cisco IOS RIPng ではコスト 0 として処理されます。ルートが Cisco NX-OS RIPng でアドバタイズされた場合、受信デバイスはすべての受信ルートに最小コストの +1 を加えた上で、ルーティング テーブルにルートをインストールします。Cisco IOS RIPng では、このコストの追加は送信側ルータで実行されるので、受信側ルータは変更なしでルートをインストールします。Cisco NX-OS および Cisco IOS デバイスの両方が連携しているときに、この動作の違いにより問題が発生する可能性があります。これらの互換性の問題は、Cisco NX-OS RIPng を、ルートのアドバタイズと処理の点で Cisco IOS RIPng と同様に動作するように構成することによって回避できます。

始める前に

RIPng を有効にする必要があります（RIPng の有効化 （512 ページ） セクションを参照）。

手順の概要

1. **configure terminal**
2. **router rip** *instance-tag*
3. **[no] metric direct 0**
4. （任意） **show running-config rip**
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**router rip** *instance-tag*
	例：
	`switch(config)# router rip 100`
	`switch(config-router)#`
		インスタンス タグを構成して、新しい RIPng インスタンスを作成します。インスタンス タグには 100、201、または 20 文字までの英数字を入力できます。

Step 3	[**no**] **metric direct 0** 例： `switch(config-router)# metric direct 0`	ルートのアドバタイズと処理の方法で Cisco IOS RIPng と Cisco NX-OS RIPng が互換性を持つようにするには、直接接続するルータすべてで、デフォルトであるコスト 1 の代わりにコスト 0 で構成します。 （注） このコマンドは、Cisco IOS デバイスを含む RIPng ネットワークに存在するすべての Cisco NX-OS デバイスで構成する必要があります。
Step 4	（任意） **show running-config rip** 例： `switch(config-router)# show running-config rip`	現在実行中の RIPng 構成を表示します。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config-router)# copy running-config startup-config`	この設定変更を保存します。

例

次に、すべての直接ルートをコスト 0 からコスト 1 に戻すことによって、Cisco IOS RIPng と Cisco NX-OS RIPng の互換性を無効化にする例を示します。

```
switch# configure terminal
switch(config)# router rip 100
switch(config-router)# no metric direct 0
switch(config-router)# show running-config rip
switch(config-router)# copy running-config startup-config
```

仮想化の設定

複数の RIPng インスタンスを構成し、複数の VRF を作成し、VRF と同数の RIPng インスタンス、または各 VRF で複数の RIPng インスタンスを使用することができます。VRF には RIPng インターフェイスを割り当てます。

（注） インターフェイスの VRF を設定した後に、インターフェイスの他のすべてのパラメータを設定します。インターフェイスの VRF を設定すると、そのインターフェイスのすべての設定が削除されます。

始める前に

RIPng を有効にする必要があります（RIPng の有効化 （512 ページ） セクションを参照）。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **exit**
4. **router rip** *instance-tag*
5. **vrf** *vrf-name*
6. （任意） **address-family ipv6 unicast**
7. **interface ethernet** *slot/port*
8. **vrf member** *vrf-name*
9. **ipv6 address** *ipv6-prefix/length*
10. **ipv6 router rip** *instance-tag*
11. （任意） **show ipv6 rip** [**instance** *instance-tag*] **interface** [*interface-type slot/port*] [**vrf** *vrf-name*]
12. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context RemoteOfficeVRF`
	`switch(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。

	コマンドまたはアクション	目的

Step 3	**exit**
	例：
	`switch(config-vrf)# exit`
	`switch(config)#`
		VRF設定モードを終了します。

Step 4	**router rip** *instance-tag*
	例：
	`switch(config)# router rip Enterprise`
	`switch(config-router)#`
		インスタンスタグを構成して、新しいRIPngインスタンスを作成します。

Step 5	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf RemoteOfficeVRF`
	`switch(config-router-vrf)#`
		新しい VRF を作成します。

Step 6	（任意） **address-family ipv6 unicast**
	例：
	`switch(config-router-vrf)# address-family ipv6 unicast`
	`switch(config-router-vrf-af)#`
		この RIPng インスタンスの VRF アドレス ファミリを構成します。

Step 7	**interface ethernet** *slot*/*port*
	例：
	`switch(config-router-vrf-af)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 8	**vrf member** *vrf-name* 例： `switch(config-if)# vrf member RemoteOfficeVRF`	このインターフェイスを VRF に追加します。
Step 9	**ipv6 address** *ipv6-prefix*/*length* 例： `switch(config-if)# ipv6 address 1001::1/64`	このインターフェイスのIPアドレスを設定します。このステップは、このインターフェイスを VRF に割り当てたあとに行う必要があります。
Step 10	**ipv6 router rip** *instance-tag* 例： `switch(config-if)# ipv6 router rip Enterprise`	このインターフェイスをRIPngインスタンスと関連付けます。
Step 11	（任意） **show ipv6 rip** [**instance** *instance-tag*] **interface** [*interface-type slot*/*port*] [**vrf** *vrf-name*] 例： `switch(config-if)# show ipv6 rip Enterprise ethernet 1/2`	VRF のインターフェイスに関するRIPng情報を表示します。

	コマンドまたはアクション	目的
Step 12	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

例

次に、VRF を作成して、その VRF にインターフェイスを追加する例を示します。

```
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 ipv6 unicast
switch(config-router-vrf-af)# interface ethernet 1/2
switch(config-if)# vrf member RemoteOfficeVRF
switch(config-if)# ipv6 address 1001::1/64
switch(config-if)# ipv6 router rip Enterprise
switch(config-if)# copy running-config startup-config
```

RIPng のチューニング

ネットワーク要件に適合するように RIPng を調整できます。RIPng では複数のタイマーを使用して、ルーティング アップデート間隔、ルートが無効になるまでの時間の長さ、およびその他のパラメータを決定します。これらのタイマーを調整すると、インターネットワークのニーズに適合するように、ルーティング プロトコルのパフォーマンスを調整できます。

（注） ネットワーク上のすべての RIPng 有効化ルータで、RIPng タイマーに同じ値を構成する必要があります。

RIPng を調整するには、アドレス ファミリ構成モードで次のオプション コマンドを使用します：

コマンド	目的
timers basic *update timeout holddown garbage-collection* 例： `switch(config-router-af)# timers basic 40 120 120 100`	RIPng タイマーを秒数で設定します。パラメータは次のとおりです。 • *update*：指定できる範囲は 5 ～任意の正の整数。デフォルトは 30 です。 • *timeout*：ルートの無効を宣言するまでに、Cisco NX-OS が待機する時間。タイムアウト インターバルが終了するまでに、このルートのアップデート情報を Cisco NX-OS が受信しなかった場合、Cisco NX-OS はルートの無効を宣言します。指定できる範囲は 1 ～任意の正の整数です。デフォルトは 180 です。 • *holddown*：無効ルートに関するよりよいルート情報を Cisco NX-OS が無視する時間。指定できる範囲は 0 ～任意の正の整数です。デフォルトは 180 です。 • *garbage-collection*：Cisco NX-OS がルートを無効として表示してから、Cisco NX-OS がそのルートをルーティング テーブルから削除するまでの時間。指定できる範囲は 1 ～任意の正の整数です。デフォルトは 120 です。

RIPng を調整するには、インターフェイス構成モードで次のオプション コマンドを使用します：

コマンド	目的
ipv6 rip route-filter {**prefix-list** *list-name* | **route-map** *map-name* | [**in** | **out**] 例： `switch(config-if)# ipv6 rip route-filter route-map InputMap in`	着信または発信 ipv6 rip アップデートをフィルタ処理するための、ルート マップを指定します。

RIPng 構成の確認

RIPng の構成を表示するには、次のいずれかの作業を行います。

コマンド	目的
show ipv6 rip instance [*instance-tag*] [**vrf** *vrf-name*]	RIPng インスタンスの状態を表示します。

コマンド	目的
show ipv6 rip [**instance** *instance-tag*] **interface** *slot/port* **detail** [**vrf** *vrf-name*]	インターフェイスの RIP ステータスを表示します。
show ipv6 rip [**instance** *instance-tag*] **neighbor** [*interface-type number*] [**vrf** *vrf-name*]	RIPng ネイバー テーブルを表示します。
show ipv6 rip [**instance** *instance-tag*] **route** [*ip-prefix/length* [**longer-prefixes** | **shorter-prefixes**]] [**summary**] [**vrf** *vrf-name*]	RIPng ルート テーブルを表示します。
show running-configuration rip	現在実行中の RIPng 構成を表示します。

RIPng 統計の表示

RIPng の統計情報を表示するには、次のコマンドを使用します。

コマンド	目的
show ipv6 rip [**instance** *instance-tag*] **statistics** *interface-type number* [**vrf** *vrf-name*]	RIPng 統計を表示します。

clear ipv6 rip statistics コマンドを使用し、 RIPng 統計情報をクリアするコマンド。

RIPng の設定例

VRF で Enterprise RIPng インスタンスを作成し、その RIPng インスタンスにイーサネット インターフェイス 1/2 を追加する例を示します

```
router rip Enterprise
address-family ipv6 unicast
distance 33
maximum-paths 8
default-information originate always
timers basic 31 181 181 121

interface ethernet 1/2
ipv6 router rip Enterprise
```

関連項目

ルートマップの詳細については、Route Policy Manager の設定（587 ページ）を参照してください。

関連項目

第 14 章

スタティック ルーティングの設定

この章では、Cisco NX-OS デバイス上でスタティック ルーティングを設定する方法について説明します。

この章は、次の内容で構成されています。

- スタティック ルーティングについて （525 ページ）
- スタティック ルーティングの前提条件 （527 ページ）
- デフォルト設定 （527 ページ）
- スタティック ルーティングの設定 （528 ページ）
- スタティック ルーティングの設定例 （533 ページ）

スタティック ルーティングについて

ルータは、ユーザが手動で設定したルートテーブルエントリのルート情報を使用するか、またはダイナミックルーティングアルゴリズムで計算されたルート情報を使用して、パケットを転送します。

スタティックルートは、2つのルータ間の明示パスを定義するものであり、自動的にはアップデートできません。ネットワークに変更が生じたときは、手動で再設定する必要があります。スタティックルートは、ダイナミックルートに比べて使用する帯域幅が少なくなります。ルーティング アップデートの計算や分析に CPU サイクルを使用しません。

必要に応じて、スタティックルートでダイナミックルートを補うことができます。スタティックルートをダイナミック ルーティング アルゴリズムに再配布できますが、ダイナミック ルーティング アルゴリズムで計算されたルーティング情報をスタティック ルーティング テーブルに再配布できません。

スタティックルートは、ネットワークトラフィックが予測可能で、ネットワーク設計が単純な環境で使用します。スタティック ルートはネットワークの変化に対応できないので、大規模でたえず変化しているネットワークでは、スタティック ルートを使用すべきではありません。大部分のネットワークは、ルータ間の通信にダイナミックルートを使用しますが、特殊な状況でスタティック ルートを 1 つか 2 つ設定する場合があります。スタティック ルートは、最終手段としてのゲートウェイ（ルーティング不能なすべてのパケットの送信先となるデフォルト ルータ）を指定する場合にも便利です。

アドミニストレーティブ ディスタンス

アドミニストレーティブ ディスタンスは、2 つの異なるルーティング プロトコルから同じ宛先に、2 つ以上のルートが存在する場合に、最適なパスを選択するために、ルータが使用するメトリックです。複数のプロトコルがユニキャストルーティングテーブルに同じルートを追加した場合に、アドミニストレーティブディスタンスを手がかりに、他のルーティングプロトコル（またはスタティック ルート）ではなく、特定のルーティング プロトコル（またはスタティック ルート）が選択されます。各ルーティングプロトコルは、アドミニストレーティブディスタンス値を使用して、信頼性の高い順にプライオリティが与えられます。

スタティック ルートのデフォルトのアドミニストレーティブ ディスタンスは 1 です。ルータは値の小さいルートが最短であると見なすので、スタティックルートがダイナミックルートより優先されます。ダイナミック ルートでスタティック ルートを上書きする場合は、スタティック ルートにアドミニストレーティブ ディスタンスを指定します。たとえば、アドミニストレーティブ ディスタンスが 120 のダイナミック ルートが 2 つある場合に、ダイナミック ルートでスタティック ルートを上書きするには、スタティックルートに 120 より大きいアドミニストレーティブディスタンスを指定します。

直接接続のスタティック ルート

直接接続のスタティック ルートでは、出力インターフェイス（あらゆるパケットを宛先ネットワークに送り出すインターフェイス）のみを指定する必要があります。ルータは宛先が出力インターフェイスに直接接続されているものと見なし、パケットの宛先をネクストホップ アドレスとして使用します。ネクストホップは、ポイントツーポイント インターフェイスの場合に限り、インターフェイスにできます。ブロードキャスト インターフェイスの場合は、ネクストホップを IPv4/IPv6 アドレスにする必要があります。

完全指定のスタティック ルート

完全指定のスタティック ルートでは、出力インターフェイス（あらゆるパケットを宛先ネットワークに送り出すインターフェイス）またはネクストホップアドレスのどちらかを指定する必要があります。完全指定のスタティック ルートを使用できるのは、出力インターフェイスがマルチアクセスインターフェイスで、ネクストホップアドレスを特定する必要がある場合です。ネクストホップ アドレスは、指定された出力インターフェイスに直接接続する必要があります。

フローティング スタティック ルート

フローティング スタティック ルートは、ダイナミック ルートをバックアップするためにルータが使用するスタティック ルートです。フローティング スタティック ルートには、バックアップするダイナミックルートより大きいアドミニストレーティブディスタンスを設定する必要があります。この場合、ルータはフローティング スタティック ルートよりダイナミック ルートを優先させます。フローティング スタティック ルートは、ダイナミック ルートが失われた場合の代用として使用できます。

（注） デフォルトでは、ルータはダイナミック ルートよりスタティック ルートを優先させます。スタティック ルートの方がダイナミック ルートより、アドミニストレーティブ ディスタンスが小さいからです。

スタティック ルートのリモート ネクスト ホップ

リモート（非直接接続）ネクストホップを指定したスタティックルートの場合、ルータに直接接続されていない隣接ルータのネクスト ホップ アドレスを指定できます。データ転送時に、スタティック ルートにリモート ネクストホップがあると、そのネクスト ホップがユニキャスト ルーティングテーブルで繰り返し使用され、リモートネクストホップに到達可能な、対応する直接接続のネクストホップ（複数可）が特定されます。

BFD

この機能では、双方向フォワーディング検出（BFD）をサポートします。BFD は、転送パスの障害を高速で検出することを目的にした検出プロトコルです。BFD は 2 台の隣接デバイス間のサブセカンド障害を検出し、BFD の負荷の一部を、サポートされるモジュール上のデータ プレーンに分散できるため、プロトコル hello メッセージよりも CPU を使いません。詳細については、『Cisco Nexus 9000 シリーズ NX-OS インターフェイスリリース 9.3(x) 設定ガイド』を参照してください。

仮想化のサポート

スタティック ルートは、仮想ルーティングおよび転送（VRF）インスタンスをサポートしています。

スタティック ルーティングの前提条件

スタティック ルーティングの前提条件は、次のとおりです。

- ネクスト ホップ アドレスを含むユニキャスト ルートがない場合、静的ルートはユニキャスト ルーティング テーブルに追加されません。

デフォルト設定

表にスタティック ルーティング パラメータのデフォルト設定を示します。

表 *25*：デフォルトのスタティック ルーティング パラメータ

パラメータ	デフォルト
アドミニストレーティブ ディスタンス	1
RIP 機能	無効化

スタティック ルーティングの設定

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

スタティック ルーティングの設定

デバイスにスタティックルートを設定できます。

手順の概要

1. **configure terminal**
2. 次のいずれかのコマンドを入力します。
   - **ip route** {*ip-prefix* | *ip-addr*/*ip-mask*} {[*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*]} [**name** *nexthop-name*] [**tag** *tag-value*] [*preference*]
   - **ipv6 route** *ipv6-prefix* {*nh-prefix* | *link-local-nh-prefix*} | {*nexthop* [*interface*] | *link-local-nexthop* [*interface*]} [**name** *nexthop-name*] [**tag** *tag-value*] [*preference*]
3. （任意） **show** {**ip** | **ipv6**} **static-route**
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

	コマンドまたはアクション	目的

Step 2	次のいずれかのコマンドを入力します。
	• **ip route** {*ip-prefix* | *ip-addr*/*ip-mask*} {[*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*]} [**name** *nexthop-name*] [**tag** *tag-value*] [*preference*]
	• **ipv6 route** *ipv6-prefix* {*nh-prefix* | *link-local-nh-prefix*} | {*nexthop* [*interface*] | *link-local-nexthop* [*interface*]} [**name** *nexthop-name*] [**tag** *tag-value*] [*preference*]
	例：
	`switch(config)# ip route 192.0.2.0/8 ethernet 1/2 192.0.2.4`
	`switch(config)# ipv6 route 2001:0DB8::/48 6::6 ethernet 2/1`
		スタティック ルートおよびこのスタティック ルート用のインターフェイスを設定します。サポートされているインターフェイスのリストを表示するには、**?** を使用します。**null 0** を使用すると、ヌル インターフェイスを指定できます。
		任意でネクスト ホップ アドレスを設定できます。
		preference 値でアドミニストレーティブ ディスタンスを設定します。範囲は 1 ～ 255 です。デフォルトは 1 です。
		（注）
		no {**ip** | **ipv6**} **route** コマンドを使用すれば、スタティック ルートを削除できます。

Step 3	（任意） **show** {**ip** | **ipv6**} **static-route** 例： `switch(config)# show ip static-route`	スタティック ルート情報を表示します。
Step 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、ヌル インターフェイスのスタティック ルートを設定する例を示します。

```
switch# configure terminal
switch(config)# ip route 1.1.1.1/32 null 0
switch(config)# copy running-config startup-config
```

VLAN を介したスタティック ルートの設定

スタティック ルートは、VLAN を介したネクスト ホップのサポートなしで設定できます。

始める前に

アクセス ポートが VLAN の一部であることを確認します。

手順の概要

1. **configure terminal**
2. **feature interface vlan**
3. **interface-vlan** *vlan-id*

4. **ip address** *ip-addr*/*length*
5. [**no**] **ip route** *ip-addr*/*length vlan-id*
6. （任意） **show ip route**
7. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**feature interface vlan** 例： `switch(config)# feature interface-vlan`	VLAN インターフェイス モードをイネーブルにします。
Step 3	**interface-vlan** *vlan-id* 例： `switch(config)# interface-vlan 10`	SVI を作成して、インターフェイス設定モードを開始します。 **vlan-id** 引数の範囲は 1 ～ 4094 です。ただし、内部スイッチ用に予約されている VLAN は除きます。
Step 4	**ip address** *ip-addr*/*length* 例： `switch(config)# ip address 192.0.2.1/8`	VLAN の IP アドレスを設定します。
Step 5	[**no**] **ip route** *ip-addr*/*length vlan-id* 例： `switch(config)# ip route 209.165.200.224/27 vlan 10`	スイッチ仮想インターフェイス（SVI）上のネクストホップなしでインターフェイスのスタティック ルートを追加します。 IP アドレスは、スイッチに接続されたインターフェイスで設定されるアドレスです。 スタティック ルートを削除するには、**no** キーワードを使用します。
Step 6	（任意） **show ip route** 例： `switch(config)# show ip route`	Unicast Route Information Base （URIB） からルートを表示します。
Step 7	（任意） **copy running-config startup-config** 例：	この設定変更を保存します。

	コマンドまたはアクション	目的
	`switch(config)# copy running-config startup-config`	

例

次に、SVIを介したネクストホップなしでスタティックルートを設定する例を示します。

```
switch# configure terminal
switch(config)# feature interface-vlan
swicth(config)# interface vlan 10
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# ip route 209.165.200.224/27 vlan 10  <===209,165.200.224 is the IP
address of the interface that is configured on the interface that is directly connected to
the switch.
switch(config-if)# copy running-config startup-config
```

仮想化の設定

VRF でスタティック ルートを設定できます。

（注） VRF コンテキストに **ip route** コマンドを適用すると、**show run vrf** コマンドにより初期設定から変更されたオクテットが表示されます。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf-name*
3. 次のいずれかのコマンドを入力します。
   - **ip route** {*ip-prefix* | *ip-addr ip-mask*} {*next-hop* | *nh-prefix* | *interface*} [**name** *nexthop-name*] [**tag** *tag-value*] [*preference*]
   - **ipv6 route** *ipv6-prefix* {*nh-prefix* | *link-local-nh-prefix*} | {*nexthop* [*interface*] | *link-local-nexthop* [*interface*]} [**name** *nexthop-name*] [**tag** *tag-value*] [*preference*]
4. （任意） **show** {**ip** | **ipv6**} **static-route vrf** *vrf-name*
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例:	グローバルコンフィギュレーションモードを開始します。

	コマンドまたはアクション	目的
	`switch# configure terminal` `switch(config)#`	

Step 2	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context StaticVrf`
	`switch(config-vrf)#`
		VRF を作成し、VRF設定モードを開始します。

Step 3	次のいずれかのコマンドを入力します。
	• **ip route** {*ip-prefix* | *ip-addr ip-mask*} {*next-hop* | *nh-prefix* | *interface*} [**name** *nexthop-name*] [**tag** *tag-value*] [*preference*]
	• **ipv6 route** *ipv6-prefix* {*nh-prefix* | *link-local-nh-prefix*} | {*nexthop* [*interface*] | *link-local-nexthop* [*interface*]} [**name** *nexthop-name*] [**tag** *tag-value*] [*preference*]
	例：
	`switch(config-vrf)# ip route 192.0.2.0/8 ethernet 1/2`
	`switch(config-vrf)# ipv6 route 2001:0DB8::/48 6::6 ethernet 2/1`
		スタティック ルートおよびこのスタティック ルート用のインターフェイスを設定します。サポートされているインターフェイスのリストを表示するには、**?** を使用します。**null 0** を使用すると、ヌル インターフェイスを指定できます。
		任意でネクスト ホップ アドレスを設定できます。
		preference 値でアドミニストレーティブ ディスタンスを設定します。範囲は 1 ～ 255 です。デフォルトは 1 です。

Step 4	（任意） **show** {**ip** | **ipv6**} **static-route vrf** *vrf-name* 例： `switch(config-vrf)# show ip static-route`	スタティック ルート情報を表示します。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config-vrf)# copy running-config startup-config`	この設定変更を保存します。

例

スタティック ルートの設定例を示します。

```
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
```

スタティック ルーティングの設定確認

スタティック ルーティングの設定を表示するには、次のいずれかの作業を行います。

コマンド	目的
show {**ip** | **ipv6**} **static-route**	設定されているスタティック ルートを表示します。
show ipv6 static-route vrf *vrf-name*	各 VRF のスタティック ルートの情報を表示します。
show {**ip** | **ipv6**} **static-route track-table**	IPv4 または IPv6 スタティック ルート トラック テーブルに関する情報を表示します。

スタティック ルーティングの設定例

次に、スタティック ルーティングの設定例を示します。

```
configure terminal
ip route 192.0.2.0/8 192.0.2.10
copy running-config startup-config
```

第 15 章

レイヤ 3 仮想化の設定

この章では、Cisco NX-OS デバイスでレイヤ 3 仮想化を設定する方法について説明します。

この章は、次の項で構成されています。

- レイヤ 3 仮想化について （535 ページ）
- VRF の前提条件 （540 ページ）
- VRF の注意事項および制約事項 （540 ページ）
- VRF ルート リークの注意事項と制約事項 （541 ページ）
- デフォルト設定 （542 ページ）
- VRF の設定 （542 ページ）
- VRF の設定の確認 （550 ページ）
- VRF の設定例 （550 ページ）
- その他の参考資料 （557 ページ）

レイヤ 3 仮想化について

Cisco NX-OS は、複数の仮想ルーティングおよび転送（VRF）インスタンスをサポートしています。各 VRF には、IPv4 および IPv6 に対応するユニキャストおよびマルチキャスト ルート テーブルを備えた、独立したアドレス空間が 1 つずつあり、他の VRF と無関係にルーティングを決定できます。

ルータごとに、デフォルト VRF および管理 VRF があります。

管理 VRF

- 管理 VRF は管理専用です。
- mgmt 0 インターフェイスのみが、管理 VRF にいることができます。
- mgmt 0 インターフェイスは、異なる VRF に割り当てられることはできません。
- ルーティング プロトコルは、管理 VRF（スタティックのみ）で動作できません。

デフォルト VRF

- すべてのレイヤ 3 インターフェイスは、別の VRF に割り当てられるまでデフォルト VRF に存在します。
- 異なる VRF コンテキストが指定されない限り、ルーティング プロトコルはデフォルトの VRF コンテキストで実行されます。
- デフォルト VRF は、すべての show コマンドに対してデフォルトのルーティング コンテキストを使用します コマンドにも表示されません。
- デフォルト VRF は、Cisco IOS のグローバル ルーティング テーブルの概念に似ています。

（注） Cisco NX-OS リリース 10.4(3)F にアップグレードすると、limit-resource のデフォルト構成が 4096 に変更されます。

Cisco NX-OS リリース 10.3(5) にアップグレードすると、limit-resource のデフォルト構成が 4096 に変更されます。

出力ロードバランス解決 VRF

出力ロードバランス解決（egress-loadbalance- Resolution-）は、自動的に作成される内部 VRF です。この VRF はデフォルト VRF に似ています。

この VRF の目的は、VXLAN EVPN 機能のルートの追加の計算と解決をサポートすることです。

（注）
- この VRF は無害な空の VRF であるため、ユーザーは構成できず、削除できません。
- VRF 制限はこの新しい暗黙的な VRF に対応するために 4096 から 4097 に引き上げられました。

  次に例を示します。

  - 既存のデフォルト構成

    ```
    vdc switch id 1
    limit-resource vrf minimum 2 maximum 4096
    ```

  - 新規デフォルト構成

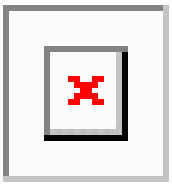
    ```
    vdc switch id 1
    limit-resource vrf minimum 2 maximum 4097
    ```

VRF およびルーティング

すべてのユニキャストおよびマルチキャスト ルーティング プロトコルは VRF をサポートします。VRF でルーティング プロトコルを設定する場合は、同じルーティング プロトコル インスタンスの別の VRF のルーティング パラメータに依存しないルーティング パラメータをその VRF に設定します。

VRF にインターフェイスおよびルーティング プロトコルを割り当てることによって、仮想レイヤ 3 ネットワークを作成できます。インターフェイスが存在する VRF は 1 つだけです。次の図は、1 つの物理ネットワークが 2 つの VRF からなる 2 つの仮想ネットワークに分割されている例を示しています。ルータ Z、A、および B は、VRF Red にあり、1 つのアドレス ドメインを形成しています。これらのルータは、Router C が含まれないルート更新を共有します。Router C は別の VRF で設定されているからです。

図 *38 :* ネットワーク内の *VRF*



デフォルトで、着信インターフェイスの VRF を使用して、ルート検索に使用するルーティング テーブルを選択します。ルート ポリシーを設定すると、この動作を変更し、Cisco NX-OS が着信パケットに使用する VRF を設定できます。

Cisco NX-OS は VRF 間のルート リーク（インポートまたはエクスポート）をサポートします。

デフォルトの VRF からのルート リークとルートのインポート

Cisco NX-OS は VRF 間のルート リーク（インポートまたはエクスポート）をサポートします。

インポートポリシーを使用して、グローバルルーティングテーブル（デフォルトVRF）から他のVRFにIPプレフィックスをインポートできます。VRF インポート ポリシーはルート マップを使用して、VRF にインポートされるプレフィックスを指定します。ポリシーは、IPv4およびIPv6ユニキャストプレフィックスをインポートできます。

（注） BGPデフォルトVRFのルートは直接インポートできます。デフォルトVRFの他のルートは、最初にBGPに再配布する必要があります。

IP プレフィックスは、標準のルート ポリシー フィルタリング メカニズムでインポート ルート マップの一致基準として定義されます。たとえば、IPプレフィックスリストまたはas-pathフィルタを作成してIPプレフィックスまたはIPプレフィックス範囲を定義し、そのプレフィックスリストまたはas-pathフィルタをルートマップのmatch句で使用できます。ルートマップを通過したプレフィックスは、インポートポリシーを使用して指定された VRF にインポートされます。このインポートポリシーによってVRFにインポートされた IP プレフィックスは、別の VRF に再インポートできません。

詳細については、「VRF ルート リーキングの注意事項および制約事項」セクションを参照してください。

IPv6 専用環境の BGP VRF ルーター ID

router-id を取得するソースを優先順に次に示します。

1. VRF レベル router-id コマンド
2. IPv4 アドレスが設定された VRF インターフェイス
3. デフォルトの VRF ルータ ID 設定からデフォルト以外の VRF ルータ ID を継承

（注） router-id の 3 番目のソースの優先度は最も低く、1 番目と 2 番目のソースが利用できない場合にのみ適用されます。

（注） router-id がない場合、BGP OPEN メッセージを送信できません。

VRF 認識サービス

Cisco NX-OS アーキテクチャの基本的な特徴として、すべての IP ベースの機能が VRF を認識することがあげられます。

次の VRF 認識サービスは、特定の VRF を選択することにより、リモート サーバへの接続や、選択した VRF に基づいた情報のフィルタリングを可能にします。

- AAA：詳細については、『Cisco Nexus 9000 Series NX-OS Security Configuration Guide』を参照してください）。
- Call Home：詳細については、『Cisco Nexus 9000 Series NX-OS System Management Configuration Guide』を参照してください。
- DNS（ドメイン ネーム システム）：詳細については、DNS の設定（105 ページ）を参照してください。
- HSRP：詳細については、『Configuring HSRP』（651 ページ）を参照してください。
- HTTP：詳細については、『Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide』を参照してください。
- NTP：詳細については、『Cisco Nexus 9000 Series NX-OS System Management Configuration Guide』を参照してください。
- Ping と Traceroute：詳細については、『Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide』を参照してください。
- RADIUS：詳細については、『Cisco Nexus 9000 Series NX-OS Security Configuration Guide』を参照してください）。
- SMNP：詳細については、『Cisco Nexus 9000 Series NX-OS System Management Configuration Guide』を参照してください。

- SSH：詳細については、『Cisco Nexus 9000 Series NX-OS Security Configuration Guide』を参照してください）。
- Syslog：詳細については、『Cisco Nexus 9000 Series NX-OS System Management Configuration Guide』を参照してください。
- TACAS+：詳細については、『Cisco Nexus 9000 Series NX-OS Security Configuration Guide』を参照してください）。
- TFTP：詳細については、『Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide』を参照してください。
- VRRP（仮想ルータ冗長プロトコル）：詳細については、VRRP の設定（681 ページ）を参照してください。
- XML：詳細については、『Cisco NX-OS XML Management Interface User Guide』を参照してください。

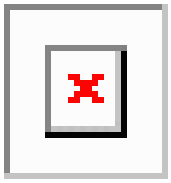
各サービスで VRF サポートを設定する詳細については、各サービスの適切なコンフィギュレーション ガイドを参照してください。

Reachability

到達可能性は、サービスを提供するサーバに到達するために必要なルーティング情報がどの VRF にあるかを示します。たとえば、管理 VRF で到達可能な SNMP サーバを設定できます。ルータにサーバ アドレスを設定する場合は、サーバに到達するために Cisco NX-OS が使用すべき VRF も設定します。

次の図は、管理 VRF を介して到達可能な SNMP サーバを示しています。SNMP サーバ ホスト 192.0.2.1 には管理 VRF を使用するように、ルータ A を設定します。

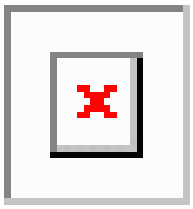
図 *39*：サービス *VRF* の到達可能性



フィルタリング

フィルタリングにより、VRF に基づいて VRF 認識サービスに渡される情報のタイプを制限できます。たとえば、Syslog サーバが特定の VRF をサポートするように設定できます。下に示す 2 つの Syslog サーバは、それぞれ 1 つの VRF をサポートしています。Syslog サーバ A は VRF Red で設定されているので、Cisco NX-OS は VRF Red で生成されたシステム メッセージだけを Syslog サーバ A に送信します。

図 *40*：サービス *VRF* のフィルタリング

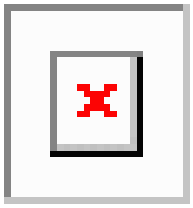


到達可能性とフィルタリングの組み合わせ

VRF 認識サービスの到達可能性とフィルタリングを組み合わせることができます。サービスに接続するために Cisco NX-OS が使用する VRF とともに、そのサービスがサポートする VRF も設定できます。デフォルト VRF でサービスを設定する場合は、任意で、すべての VRF をサポートするようにサービスを設定できます。

次の図は、管理 VRF を介して到達可能な SNMP サーバを示しています。たとえば、SNMP サーバが VRF Red からの SNMP 通知だけをサポートするように設定できます。

図 *41*：サービス *VRF* の到達可能性とフィルタリング



VRF の前提条件

デフォルト VDC 以外の仮想デバイス コンテキスト（VDC）を使用するには、Advanced Services ライセンスをインストールする必要があります。VRF のライセンス要件は VDC と同じです。

VRF の注意事項および制約事項

VRF 設定時の注意事項と制約事項は次のとおりです。

- プレフィックス リスト内の名前は、大文字と小文字が区別されません。一意の名前を使用することを推奨します。大文字と小文字を変更しただけの名前は使用しないでください。たとえば、CTCPrimaryNetworks と CtcPrimaryNetworks は 2 つの異なるエントリではありません。
- インターフェイスを既存の VRF のメンバにすると、Cisco NX-OS はあらゆるレイヤ 3 設定を削除します。VRF にインターフェイスを追加したあとで、すべてのレイヤ 3 パラメータを設定する必要があります。
- 管理 VRF に mgmt0 インターフェイスを追加し、そのあとで mgmt0 の IP アドレスおよびその他のパラメータを設定します。

- VRF が存在しないうちに VRF のインターフェイスを設定した場合は、VRF を作成するまで、そのインターフェイスは運用上のダウンになります。
- Cisco NX-OS はデフォルトで、デフォルトと管理 VRF を作成します。mgmt0 は管理 VRF のメンバにする必要があります。
- この項で説明している **write erase boot** コマンドを実行しても、管理 VRF の設定は削除されません。**write erase** を使用する必要があります コマンドを使用し、 **write erase boot** コマンドを使用する必要があります。
- ルート ターゲットには、次の注意事項と制約事項があります。
  - レイヤ 2 とレイヤ 3 に異なるルート ターゲットを割り当てるのがベスト プラクティスです。
  - 自動ルート ターゲット生成では、ルート ターゲットは EVI から生成されます。レイヤ 2 とレイヤ 3 で異なる EVI 範囲を使用して、レイヤ 2 とレイヤ 3 の EVI が同じ識別子を使用しないようにすることをお勧めします。
- Cisco NX-OS リリース 10.3(1)F 以降、マルチ VRF が Cisco Nexus 9808 プラットフォーム スイッチでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、マルチ VRF が Cisco Nexus 9804 プラットフォーム スイッチでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、マルチ VRF は、Cisco Nexus 9808 および 9804 スイッチを搭載した Cisco Nexus X98900CD-A および N9KX9836DM-A ライン カードでサポートされます。

VRF ルート リークの注意事項と制約事項

VRF ルート リークには次の設定注意事項と制限があります。

- ルート リークは、任意の 2 つのデフォルト以外の VRF間、およびデフォルト VRF からデフォルト以外の VRF にサポートされます。

（注） VRF 間のルート リークは、MPLS セグメント ルーティング（SR-MPLS）ではサポートされません。

VRF 間のルートリークは BGP ではサポートされません。BGP スピーカーは、異なる VRF を介してルーティングされるピア IP には接続できません。

- デフォルト VRF へのルート リークは、グローバル VRF であるため使用できません。
- 指定した IP アドレスにマッチするルートマップのフィルタを使用して、特定のルートに対してルート リークを制限できます。

- ルートが別の VRF を指している場合、NX-OS と Guestshell コンテナ間のルート同期は行われません。
- デフォルトでは、デフォルト VRF からデフォルト以外の VRF にインポートできる（逆も可能） IP プレフィックスの最大数は 1000 ルートです。
- 2 つの非デフォルト VRF間でリークできるルートの数に制限はありません。
- Cisco NX-OS リリース 10.3(1)F 以降、VRF 間のルート リークは Cisco Nexus 9808 スイッチでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、VRF 間のルート リークは Cisco Nexus 9804 スイッチでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、VRF 間のルートリークは、Cisco Nexus 9808 および 9804 スイッチを搭載した Cisco Nexus X98900CD-A および X9836DM-A ライン カードでサポートされます。

デフォルト設定

次の表に、VRF パラメータのデフォルト設定値を示します。

表 *26 :* デフォルトの *VRF* パラメータ

パラメータ	デフォルト
設定されている VRF	デフォルト、管理
ルーティング コンテキスト	デフォルト VRF

VRF の設定

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

VRF の作成

VRF を作成できます。

（注） グローバル設定モードで使用できるコマンドはすべて、VRF 設定モードでも使用できます。

手順の概要

1. **configure terminal**
2. [**no**] **vrf context** *name*
3. （任意） **ip route** {*ip-prefix* | *ip-addr ip-mask*} {[*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*]} [**tag** *tag-value* [*preference*]
4. （任意） **show vrf** [*vrf-name*]
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	[**no**] **vrf context** *name*
	例：
	`switch(config)# vrf context Enterprise`
	`switch(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。*name* には最大 32 文字の英数字を使用できます。大文字と小文字は区別されます。
		このコマンドで **no** オプションを使用すると、VRF と、関連するすべての設定が削除されます。

Step 3	（任意） **ip route** {*ip-prefix* | *ip-addr ip-mask*} {[*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*]} [**tag** *tag-value* [*preference*] 例： `switch(config-vrf)# ip route 192.0.2.0/8 ethernet 1/2 192.0.2.4`	スタティック ルートおよびこのスタティック ルート用のインターフェイスを設定します。任意でネクスト ホップ アドレスを設定できます。*preference* 値でアドミニストレーティブディスタンスを設定します。範囲は 1 ～ 255 です。デフォルトは 1 です。
Step 4	（任意） **show vrf** [*vrf-name*] 例： `switch(config-vrf)# show vrf Enterprise`	VRF 情報を表示します。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config-vrf)# copy running-config startup-config`	この設定変更を保存します。

例

次に、VRF を作成し、VRF にスタティック ルートを追加する例を示します。

```
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
```

インターフェイスへの VRF メンバーシップの割当て

インターフェイスを VRF のメンバにできます。

始める前に

VRF 用のインターフェイスを設定したあとで、インターフェイスにIPアドレスを割り当てます。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrf member** *vrf-name*
4. **ip address** *ip-prefix/length*
5. （任意） **show vrf** *vrf-name* **interface** *interface-type number*
6. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**vrf member** *vrf-name* 例： `switch(config-if)# vrf member RemoteOfficeVRF`	このインターフェイスを VRF に追加します。

	コマンドまたはアクション	目的
Step 4	**ip address** *ip-prefix/length* 例： `switch(config-if)# ip address 192.0.2.1/16`	このインターフェイスの IP アドレスを設定します。このステップは、このインターフェイスを VRF に割り当てたあとに行う必要があります。
Step 5	（任意） **show vrf** *vrf-name* **interface** *interface-type number* 例： `switch(config-vrf)# show vrf Enterprise interface ethernet 1/2`	VRF 情報を表示します。
Step 6	（任意） **copy running-config startup-config** 例： `switch(config-vrf)# copy running-config startup-config`	この設定変更を保存します。

例

次に、VRF にインターフェイスを追加する例を示します。

```
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
```

ルーティング プロトコル用の VRF パラメータの設定

1 つまたは複数の VRF にルーティング プロトコルを関連付けることができます。ルーティング プロトコルに関する VRF の設定については、該当する章を参照してください。ここでは、詳細な設定手順の例として、OSPFv2 プロトコルを使用します。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **vrf** *vrf-name*
4. （任意） **maximum-paths** *paths*
5. **exit**
6. **exit**
7. **interface** *interface-type slot/port*
8. **vrf member** *vrf-name*
9. **ip address** *ip-prefix/length*
10. **ip router ospf** *instance-tag* **area** *area-id*

11. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**router ospf** *instance-tag*
	例：
	`switch (config-vrf)# router ospf 201`
	`switch(config-router)#`
		インスタンスタグが設定された新しい OSFPv2 インスタンスを作成します。

Step 3	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf RemoteOfficeVRF`
	`switch(config-router-vrf)#`
		VRF 設定モードを開始します。

Step 4	（任意） **maximum-paths** *paths* 例： `switch(config-router-vrf)# maximum-paths 4`	この VRF のルート テーブル内の宛先への、同じ OSPFv2 パスの最大数を設定します。このコマンドはロード バランシングに使用されます。

Step 5	**exit**
	例：
	`switch(config-router-vrf)# exit`
	`switch(config-router)#`
		VRF設定モードを終了します。

Step 6	**exit**
	例：
	`switch(config-router)# exit`
	`switch(config)#`
		ルータ設定モードを終了します。

Step 7	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 8	**vrf member** *vrf-name* 例：	このインターフェイスを VRF に追加します。

	コマンドまたはアクション	目的
	`switch(config-if)# vrf member RemoteOfficeVRF`	
Step 9	**ip address** *ip-prefix/length* 例： `switch(config-if)# ip address 192.0.2.1/16`	このインターフェイスのIPアドレスを設定します。このステップは、このインターフェイスを VRF に割り当てたあとに行う必要があります。
Step 10	**ip router ospf** *instance-tag* **area** *area-id* 例： `switch(config-if)# ip router ospf 201 area 0`	このインターフェイスを OSPFv2 インスタンスおよび設定エリアに割り当てます。
Step 11	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップコンフィギュレーションにコピーします

例

次に、VRF を作成して、その VRF にインターフェイスを追加する例を示します。

```
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
```

VRF 認識サービスの設定

VRF 認識サービスの到達可能性とフィルタリングを設定できます。

ここでは、サービスの詳細な設定手順の例として、SNMP および IP ドメイン リストを使用します。

手順の概要

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. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	**snmp-server host** *ip-address* [**filter-vrf** *vrf-name*] [**use-vrf** *vrf-name*] 例： `switch(config)# snmp-server host 192.0.2.1 use-vrf Red`	グローバル SNMP サーバを設定し、サービスに到達するために Cisco NX-OS が使用する VRF を設定します。選択した VRF からこのサーバへの情報をフィルタリングするには、**filter-vrf** キーワードを使用します。

Step 3	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context Blue`
	`switch(config-vrf)#`
		新しい VRF を作成します。

Step 4	**ip domain-list** *domain-name* [**all-vrfs**] [**use-vrf** *vrf-name*] 例： `switch(config-vrf)# ip domain-list List all-vrfs use-vrf Blue`	VRF でドメイン リストを設定し、必要に応じて、リスト内のドメイン名に到達するために Cisco NX-OS が使用する VRF を設定します。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config-vrf)# copy running-config startup-config`	この設定変更を保存します。

例

次の例は、VRF Red 上の到達可能な SNMP ホスト 192.0.2.1 に、すべての VRF の SNMP 情報を送信する方法を示しています。

```
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
```

次の例は、VRF Red 上の到達可能な SNMP ホスト 192.0.2.12 に対して、VRF Blue の SNMP 情報をフィルタリングする方法を示しています。

```
switch# configure terminal
switch(config)# vrf context Blue
```

```
switch(config-vrf)# snmp-server host 192.0.2.12 use-vrf Red
switch(config)# copy running-config startup-config
```

VRF スコープの設定

すべての EXEC コマンド（**show** コマンドなど）の VRF スコープを設定できます。そうすることで、EXEC コマンド出力のスコープが設定された VRF に自動的に限定されます。このスコープは、一部の EXEC コマンドで使用できる VRF キーワードによって上書きできます。

手順の概要

1. **routing-context vrf** *vrf-name*

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**routing-context vrf** *vrf-name*
	例：
	`switch# routing-context vrf red`
	`switch%red#`
		すべての EXEC コマンドに対応するルーティング コンテキストを設定します。デフォルトのルーティング コンテキストはデフォルト VRF です。
		（注）
		routing-context vrf default コマンドを使用し、 コマンドを使用して、デフォルトの VRF スコープに戻ります。

例

デフォルトの VRF スコープに戻すには、EXEC モードで次のコマンドを使用します。

コマンド	目的

routing-context vrf default
例：
`switch%red# routing-context vrf default`
`switch#`
	デフォルトのルーティング コンテキストを設定します。

（注） BGP 構成を使用して VPN VRF をシャットダウンすると、シャットダウン プロセスが完了するまでに約 50 秒かかります。

VRF の設定の確認

VRF 設定情報を表示するには、次のいずれかの作業を実行します。

コマンド	目的
show bgp process vrf [*vrf-name*]	すべてまたは 1 つの VRF の情報を表示します。
show vrf [*vrf-name*]	すべてまたは 1 つの VRF の情報を表示します。
show vrf [*vrf-name*] **detail**	すべてまたは 1 つの VRF の詳細情報を表示します。
show vrf [*vrf-name*] [**interface** *interface-type slot/port*]	インターフェイスの VRF ステータスを表示します。

VRF の設定例

VRF 構成の詳細については、「VRF の設定 （542 ページ）」を参照してください。

（注）

- **snmp-server host** コマンドは、グローバルと **vrf-context** コマンドの両方で使用できます。以下の例では、**vrf-context** での構成例を示します。
- 属性（use-vrf、source-interface、filter-vrf など）をアタッチする前に、スイッチでホストが構成されていることを確認します。

VRF Red の構成例

```
!SNMP server configuration under VRF context Red:
vrf context Red
  snmp-server host 192.168.0.12 use-vrf Red

!OSPF instance configuration to VRF Red
router ospf 201
  vrf Red

!interface configuration for VRF Red
interface ethernet 1/2
  vrf member Red
  ip address 192.168.0.1/16
  ip router ospf 201 area 0
  no shutdown
```

VRF Red および Blue の構成例

```
!VRFs (Red, and Blue) creation
vrf context Red
vrf context Blue
```

```
!Configures OSPF per VRF
feature ospf
router ospf Lab
  vrf Red

router ospf Production
  vrf Blue
      router-id 192.168.1.0

interface ethernet 1/2
  vrf member Red
  ip address 192.168.0.1/16
  ip router ospf Lab area 0
  no shutdown

interface ethernet 10/2
  vrf member Blue
  ip address 192.168.0.1/16
  ip router ospf Production area 0
  no shutdown

!SNMP server configuration under VRF
!Note: Use the SNMP context "lab" to access the OSPF-MIB values for the OSPF instance Lab
in VRF "Red" in this example.
!Create SNMP entities (v2c and/or v3) with appropriate groups/roles as needed on the switch
 to access the MIBs on the switch
!Create SNMP v3 user that can be used for SNMP queries, for example:
snmp-server user admin network-admin auth md5 password1
!Create SNMP v2c community  that can be used for SNMP queries, for example:
snmp-server community public ro

!Create SNMP contexts that can be used along with the entities for SNMP queries, for example:
snmp-server context lab instance Lab vrf Red
snmp-server context production instance Production vrf Blue
```

ルート リークの VRF 構成例

```
!VRF configuration
feature bgp
vrf context red
  ip route 192.168.33.0/32 192.168.3.1
  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 red
  ip address 192.168.3.2/24
  no shutdown

vrf context blue
  ip route 192.168.44.0/32 192.168.4.1
  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 blue
```

```
  ip address 192.168.4.2/24
  no shutdown
!IP prefix list configuration
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 192.168.101.101/32 192.168.55.1
!BGP per VRF assignment
router bgp 100
  address-family ipv4 unicast
    redistribute static route-map test

  vrf red
    address-family ipv4 unicast
      redistribute static route-map test

 vrf blue
    address-family ipv4 unicast
      redistribute static route-map test
```

グローバル VRF と非デフォルト VRF 間のルート リークの確認例

```
switch# 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>

192.168.55.0/24, ubest/mbest: 1/0, attached
    *via 192.168.55.5, Lo0, [0/0], 00:07:59, direct
192.168.55.5/32, ubest/mbest: 1/0, attached
    *via 192.168.55.5, Lo0, [0/0], 00:07:59, local
192.168.101.101/32, ubest/mbest: 1/0
    *via 192.168.55.1, [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 "red"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

192.168.33.0/32, ubest/mbest: 1/0
    *via 192.168.3.1, [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
192.168.44.0/32, ubest/mbest: 1/0
    *via 192.168.4.1%blue, [20/0], 00:12:08, bgp-100, external, tag 100
```

```
192.168.101.101/32, ubest/mbest: 1/0
    *via 192.168.55.1%default, [20/0], 00:07:41, bgp-100, external, tag 100
!
IP Route Table for VRF "blue"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>
192.168.33.0/32, ubest/mbest: 1/0
    *via 192.168.3.1%red, [20/0], 00:12:34, bgp-100, external, tag 100
192.168.44.0/32, ubest/mbest: 1/0
    *via 192.168.4.1, [1/0], 00:23:16, static
45.45.1.0/24, ubest/mbest: 1/0, attached
    *via 192.168.4.2, Eth1/11, [0/0], 00:25:53, direct
192.168.4.2/32, ubest/mbest: 1/0, attached
    *via 192.168.4.2, Eth1/11, [0/0], 00:25:53, local
192.168.101.101/32, ubest/mbest: 1/0
    *via 192.168.55.1%default, [20/0], 00:07:41, bgp-100, external, tag 100
switch(config)#
```

エクスポート VRF デフォルトの構成例

次に、「export vrf default」コマンドで導入されたインポート済みルートの再インポートを許可し、VPN インポート済みルートを default-VRF に再インポートできるようにする例を示します。

```
vrf context vpn1
  address-family ipv4 unicast
    export vrf default [<prefix-limit>] map <route-map> [allow-vpn]
  address-family ipv6 unicast
    export vrf default [<prefix-limit>] map <route-map> [allow-vpn]
```

Border-leaf の構成例

- IP プレフィックス リストを構成するには、次のコマンドを使用します。

```
!IP prefix list configuration
ip prefix-list DEFAULT_ROUTE seq 5 permit 0.0.0.0/0
route-map NO_DEFAULT_ROUTE deny 5
  match ip address prefix-list DEFAULT_ROUTE
route-map NO_DEFAULT_ROUTE permit 10
route-map allow permit 10

!Creation of VRFs, and importing the route maps
vrf context vni100
  vni 100
  ip route 0.0.0.0/0 Null0
  rd auto
  address-family ipv4 unicast
    route-target import 100:200
    route-target import 100:200 evpn
    route-target both auto
    route-target both auto evpn
    import vrf default map allow
    export vrf default map NO_DEFAULT_ROUTE allow-vpn

vrf context vni200
  vni 200
  ip route 0.0.0.0/0 Null0
  rd auto
  address-family ipv4 unicast
    route-target import 100:100
    route-target import 100:100 evpn
    route-target both auto
```

```
        route-target both auto evpn
        import vrf default map allow
        export vrf default map NO_DEFAULT_ROUTE

    !BGP configuration
    router bgp 100
      address-family ipv4 unicast
        redistribute direct route-map allow
      address-family ipv6 unicast
        redistribute direct route-map allow

      neighbor 192.168.101.101
        remote-as 100
        update-source loopback0
        address-family l2vpn evpn
          send-community extended

      neighbor 192.168.30.2
        remote-as 300
        address-family ipv4 unicast

      vrf vni100
        address-family ipv4 unicast
          network 0.0.0.0/0
          advertise l2vpn evpn
          redistribute direct route-map allow

      vrf vni200
        address-family ipv4 unicast
          network 0.0.0.0/0
          advertise l2vpn evpn
          redistribute direct route-map allow
```

BGP IPv4 ユニキャスト構成の確認例

```
switch(config-vrf)# show bgp ipv4 unicast 192.168.11.11/32
BGP routing table information for VRF default, address family IPv4 Unicast
BGP routing table entry for 192.168.11.11/32, version 14
Paths: (1 available, best #1)
Flags: (0x08041a) on xmit-list, is in urib, is best urib route, is in HW

  Advertised path-id 1
  Path type: internal, path is valid, is best path, in rib
             Imported from 192.168.3.3:3:192.168.11.11/32 (VRF vni100)
  AS-Path: 150 , path sourced external to AS
    192.168.1.0 (metric 81) from 192.168.101.101 (192.168.101.101)
      Origin incomplete, MED 0, localpref 100, weight 0
      Received label 100
      Extcommunity:
          RT:100:100
          ENCAP:8
          Router MAC:5254.004e.a437
      Originator: 192.168.1.0 Cluster list: 192.168.101.101

  Path-id 1 advertised to peers:
    192.168.30.2
```

VRF ごとの BGP IPv4 ユニキャスト構成の確認例

```
switch(config-vrf)# show bgp vrf vni100 ipv4 unicast 192.168.11.11/32
BGP routing table information for VRF vni100, address family IPv4 Unicast
BGP routing table entry for 192.168.11.11/32, version 8
Paths: (1 available, best #1)
```

```
Flags: (0x08041e) on xmit-list, is in urib, is best urib route, is in HW
  vpn: version 19, (0x100002) on xmit-list

  Advertised path-id 1, VPN AF advertised path-id 1
  Path type: internal, path is valid, is best path, in rib
             Imported from 192.168.1.0:3:[5]:[0]:[0]:[32]:[192.168.11.11]:[0.0.0.0]/224
  AS-Path: 150 , path sourced external to AS
    192.168.1.0 (metric 81) from 192.168.101.101 (192.168.101.101)
      Origin incomplete, MED 0, localpref 100, weight 0
      Received label 100
      Extcommunity:
          RT:100:100
          ENCAP:8
          Router MAC:5254.004e.a437
      Originator: 192.168.1.0 Cluster list: 192.168.101.101

  VRF advertise information:
  Path-id 1 not advertised to any peer

  VPN AF advertise information:
  Path-id 1 not advertised to any peer
```

BGP IPv6 ユニキャスト構成の確認例

```
switch(config-vrf)# show bgp ipv6 unicast 2001:DB8:1::1/64
BGP routing table information for VRF default, address family IPv6 Unicast
BGP routing table entry for 2001:DB8:1::1/64, version 13
Paths: (1 available, best #1)
Flags: (0x08041a) on xmit-list, is in u6rib, is best u6rib route, is in HW

  Advertised path-id 1
  Path type: internal, path is valid, is best path
             Imported from 192.168.3.3:3:2001:DB8:1::1/64 (VRF vni100)
  AS-Path: 150 , path sourced external to AS
    ::ffff:192.168.1.0 (metric 81) from 192.168.101.101 (192.168.101.101)
      Origin incomplete, MED 0, localpref 100, weight 0
      Received label 100
      Extcommunity:
          RT:100:100
          ENCAP:8
          Router MAC:5254.004e.a437
      Originator: 192.168.1.0 Cluster list: 192.168.101.101

  Path-id 1 advertised to peers:
    30::2
```

VRF ごとの BGP IPv6 ユニキャスト構成の確認例

```
switch(config-vrf)# show bgp vrf vni100 ipv6 unicast 2001:DB8:1::1/64
BGP routing table information for VRF vni100, address family IPv6 Unicast
BGP routing table entry for 2001:DB8::1/128, version 6
Paths: (1 available, best #1)
Flags: (0x08041e) on xmit-list, is in u6rib, is best u6rib route, is in HW
  vpn: version 7, (0x100002) on xmit-list

  Advertised path-id 1, VPN AF advertised path-id 1
  Path type: internal, path is valid, is best path
             Imported from 192.168.1.0:3:[5]:[0]:[0]:[128]:[2001:DB8:1::1]:[0::]/416
  AS-Path: 150 , path sourced external to AS
    ::ffff:192.168.1.0 (metric 81) from 192.168.101.101 (192.168.101.101)
      Origin incomplete, MED 0, localpref 100, weight 0
      Received label 100
      Extcommunity:
          RT:100:100
```

```
          ENCAP:8
          Router MAC:5254.004e.a437
      Originator: 192.168.1.0 Cluster list: 192.168.101.101

  VRF advertise information:
  Path-id 1 not advertised to any peer

  VPN AF advertise information:
  Path-id 1 not advertised to any peer
```

IPv4 ルート構成の確認例

```
switch(config-if)# 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>

0.0.0.0/0, ubest/mbest: 1/0
    *via vrf vni100, Null0, [20/0], 1d04h, bgp-100, external, tag 100
192.168.1.0/32, ubest/mbest: 1/0
    *via 192.168.103.1, Eth1/1, [110/81], 1d04h, ospf-100, intra
192.168.2.2/32, ubest/mbest: 1/0
    *via 192.168.103.1, Eth1/1, [110/81], 1d04h, ospf-100, intra
192.168.3.3/32, ubest/mbest: 2/0, attached
    *via 192.168.3.3, Lo0, [0/0], 1d04h, local
    *via 192.168.3.3, Lo0, [0/0], 1d04h, direct
192.168.9.9/32, ubest/mbest: 1/0, attached
    *via 192.168.9.9%vni100, Lo9, [20/0], 1d03h, bgp-100, external, tag 100
192.168.10.0/24, ubest/mbest: 1/0
    *via 192.168.1.0, [200/0], 1d04h, bgp-100, internal, tag 100 (evpn) segid: 100 tunnelid:
 0x1010101 encap: VXLAN
192.168.11.11/32, ubest/mbest: 1/0
    *via 192.168.1.0, [200/0], 1d04h, bgp-100, internal, tag 150 (evpn) segid: 100 tunnelid:
 0x1010101 encap: VXLAN
192.168.20.0/24, ubest/mbest: 1/0
    *via 192.168.2.2, [200/0], 1d04h, bgp-100, internal, tag 100 (evpn) segid: 200 tunnelid:
 0x2020202 encap: VXLAN
192.168.22.22/32, ubest/mbest: 1/0
    *via 192.168.2.2, [200/0], 1d04h, bgp-100, internal, tag 250 (evpn) segid: 200 tunnelid:
 0x2020202 encap: VXLAN
192.168.30.0/24, ubest/mbest: 1/0, attached
    *via 192.168.30.1, Eth1/2, [0/0], 1d04h, direct
192.168.30.1/32, ubest/mbest: 1/0, attached
    *via 192.168.30.1, Eth1/2, [0/0], 1d04h, local
192.168.33.0/32, ubest/mbest: 1/0
    *via 192.168.30.2, [20/0], 1d04h, bgp-100, external, tag 300
192.168.100.0/24, ubest/mbest: 1/0, attached
    *via 192.168.100.3%vni100, Vlan100, [20/0], 1d04h, bgp-100, external, tag 100
192.168.101.0/24, ubest/mbest: 1/0
    *via 192.168.103.1, Eth1/1, [110/80], 1d04h, ospf-100, intra
192.168.101.101/32, ubest/mbest: 1/0
    *via 192.168.103.1, Eth1/1, [110/41], 1d04h, ospf-100, intra
192.168.102.0/24, ubest/mbest: 1/0
    *via 192.168.103.1, Eth1/1, [110/80], 1d04h, ospf-100, intra
192.168.103.0/24, ubest/mbest: 1/0, attached
    *via 192.168.103.2, Eth1/1, [0/0], 1d04h, direct
192.168.103.2/32, ubest/mbest: 1/0, attached
```

IPv6 ルート構成の確認例

```
switch(config-if)# show ipv6 route
IPv6 Routing 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>

::/0, ubest/mbest: 1/0
    *via vrf vni100, Null0, [20/0], 1d04h, bgp-100, external, tag 100
2001:DB8:1::/64, ubest/mbest: 1/0, attached
    *via 2001:DB8:1::1, Eth1/1, [0/0], 1d04h, direct
2001:DB8:2:2::/128, ubest/mbest: 1/0
    *via 2001:DB8:103:1::1, Eth1/1, [110/81], 1d04h, ospf-100, intra
2001:DB8:3:3::/128, ubest/mbest: 2/0, attached
    *via 2001:DB8:3:3::3, Lo0, [0/0], 1d04h, local
    *via 2001:DB8:3:3::3, Lo0, [0/0], 1d04h, direct
2001:DB8:9:9::/128, ubest/mbest: 1/0, attached
    *via 2001:DB8:9:9::9%vni100, Lo9, [20/0], 1d03h, bgp-100, external, tag 100
2001:DB8:10::/64, ubest/mbest: 1/0
   *via 2001:DB8:1::, [200/0], 1d04h, bgp-100, internal, tag 100 (evpn) segid: 100 tunnelid:
 0x1010101 encap: VXLAN
2001:DB8:11:11::/128, ubest/mbest: 1/0
   *via 2001:DB8:1::, [200/0], 1d04h, bgp-100, internal, tag 150 (evpn) segid: 100 tunnelid:
 0x1010101 encap: VXLAN
2001:DB8:20::/64, ubest/mbest: 1/0
    *via 2001:DB8:2:2::2, [200/0], 1d04h, bgp-100, internal, tag 100 (evpn) segid: 200
tunnelid: 0x2020202 encap: VXLAN
2001:DB8:22:22::/128, ubest/mbest: 1/0
    *via 2001:DB8:2:2::2, [200/0], 1d04h, bgp-100, internal, tag 250 (evpn) segid: 200
tunnelid: 0x2020202 encap: VXLAN
2001:DB8:30::/64, ubest/mbest: 1/0, attached
    *via 2001:DB8:30::1, Eth1/2, [0/0], 1d04h, direct
2001:DB8:30::1/128, ubest/mbest: 1/0
```

その他の参考資料

仮想化の実装に関連する詳細情報については、次の項を参照してください。

VRF の関連資料

関連項目	マニュアル タイトル
VRF	『*Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide*』 『*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*』

標準

標準	タイトル
この機能でサポートされる新規の標準または変更された標準はありません。また、既存の標準のサポートは変更されていません。	—

標準

第 16 章

ユニキャスト RIB および FIB の管理

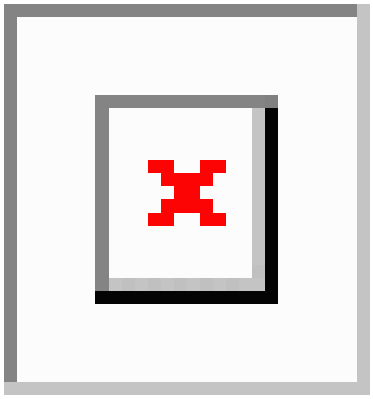
この章は、次の項で構成されています。

- ユニキャスト RIB および FIB について （559 ページ）
- ユニキャスト RIB に関する注意事項と制約事項 （560 ページ）
- ユニキャスト RIB および FIB の管理 （561 ページ）
- ユニキャスト RIB および FIB の確認 （584 ページ）
- その他の参考資料 （585 ページ）

ユニキャスト RIB および FIB について

次の図に示すように、ユニキャスト ルーティング情報ベース（IPv4 RIB および IPv6 RIB）および転送情報ベース（FIB）は、Cisco NX-OS 転送アーキテクチャの一部です。

図 *42 : CiscoNX-OS* フォワーディング アーキテクチャ



ユニキャスト RIB はアクティブなスーパーバイザ上にあります。ユニキャスト RIB は、直接接続のルート、スタティック ルート、ダイナミック ユニキャスト ルーティング プロトコルで検出されたルートを含むルーティング テーブルを維持しています。また、アドレス解決プロトコル（ARP）などの送信元から、隣接情報を収集します。ユニキャスト RIB は、ルートに最適なネクストホップを決定し、さらにユニキャスト FIB 分散モジュール（UFDM）のサービスを使用して、モジュール上のユニキャスト FIB にデータを入力します。

各ダイナミック ルーティング プロトコルは、タイム アウトしたあらゆるルートについて、ユニキャスト RIB を更新する必要があります。その後、ユニキャスト RIB はそのルートを削除し、そのルートに最適なネクスト ホップを再計算します（代わりに使用できるパスがある場合）。

レイヤ 3 整合性チェッカー

まれな事例として、各モジュールのユニキャスト RIB と FIB の間に不整合が発生することがあります。Cisco NX-OS は、レイヤ 3 整合性チェッカーをサポートします。この機能は、スーパーバイザ モジュールのユニキャスト IPv4 RIB と各インターフェイス モジュールの FIB の間で不整合を検出します。不整合には次のようなものがあります。

- 欠落したプレフィックス
- 余分なプレフィックス
- ネクストホップ アドレスの誤り
- ARP またはネイバー探索（ND）キャッシュ内の不正なレイヤ 2 リライト文字列

レイヤ 3 整合性チェッカーは、FIB のエントリと隣接マネージャ（AM）から取得した最新の隣接情報を比較し、不整合があれば記録します。次に整合性チェッカーは、ユニキャスト RIB のプレフィックスをモジュールの FIB と比較し、不整合があればログに記録します。「レイヤ 3 整合性チェッカーのトリガー」の項を参照してください。

不整合は手動で解消できます。「FIB 内の転送情報の消去」の項を参照してください。

ハードウェア制限を超えて多くのルートが学習され、**show consistency-checker forwarding ipv4** コマンドを実行した場合も、整合性の点で合格します。整合性のない状態から整合性のある状態に移行する場合も同様です。失敗として表示される場合があります。**test forwarding ipv4 inconsistency route** コマンドが再実行されるまで、この状態は終了しません。これは予期された動作です。

ユニキャスト RIB に関する注意事項と制約事項

URIB または U6RIB には、次の注意事項と制約事項が適用されます。

- 仮想ドメイン コンテキスト（VDC）では、IPv4 または IPv6 ユニキャスト ルートのメモリ リソースの制限を変更しても、変更された制限はすぐには有効になりません。

  変更された制限をアクティブにするには、**copy running-config startup-config** コマンドの後に **reload** コマンドを発行する必要があります。

  たとえば、次のいずれかのコマンドを発行した場合、新しい設定をアクティブにするには、**copy running-config startup-config** を発行し、さらにスイッチをリロードする必要があります。

  - **limit-resource u4route-mem**
  - **limit-resource u6route-mem**

（注） limit-resource に「feature pim」が構成されている場合、**limit-resource u4route-mem** プラス **limit-resource u6route-mem** の値が 1024 MB（1GB）以下であることを確認してください。

- Cisco NX-OS リリース 10.3(1)F 以降、ユニキャスト整合性チェッカは Cisco Nexus 9808 プラットフォーム スイッチでサポートされています。
  - Cisco NX-OS リリース 10.4(1)F 以降、ユニキャスト一貫性チェッカーは Cisco Nexus X98900CD-A および Cisco Nexus 9808 スイッチを搭載した X9836DM-A ライン カードでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、ユニキャスト一貫性チェッカーは Cisco Nexus 9804 プラットフォーム スイッチ、Cisco Nexus X98900CD-A および X9836DM-A ライン カードでサポートされます。
- Cisco NX-OS リリース 10.5(1)F 以降、レイヤ 3 ECMP ダイナミックロードバランシング（DLB）機能は、発信リンクの現在の使用状況に応じて、トラフィックを効率的にロードバランシングするためのサポートを提供します。

ユニキャスト RIB および FIB の管理

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

モジュールの FIB 情報の表示

モジュールの FIB 情報を表示するには、任意のモードで次のコマンドを使用します。

コマンド	目的
show forwarding {**ipv4** | **ipv6**} **adjacency module** *slot* 例： `switch# show forwarding ipv6 adjacency module 2`	IPv4 または IPv6 の隣接情報を表示します。
show forwarding {**ipv4** | **ipv6**} **route module** *slot* 例： `switch# show forwarding ipv6 route module 2`	IPv4 または IPv6 のルート テーブルを表示します。

ユニキャスト FIB でのロード シェアリングの設定

Open Shortest Path First（OSPF）などのダイナミック ルーティング プロトコルは、等コスト マルチパス（ECMP）によるロード シェアリングをサポートしています。ルーティング プロトコルは、そのプロトコルに設定されたメトリックに基づいて最適なルートを決定し、そのプロトコルに設定された最大数までのパスをユニキャスト RIB に組み込みます。ユニキャスト RIB は、RIB に含まれるすべてのルーティング プロトコル パスのアドミニストレーティブ ディスタンスを比較し、ルーティング プロトコルによって組み込まれたすべてのパス セットから最適なパス セットを選択します。ユニキャスト RIB は、この最適なパス セットを FIB に組み込み、フォワーディング プレーンで使用できるようにします。

フォワーディング プレーンは、ロード シェアリングのアルゴリズムを使用して、FIB に組み込まれたパスのいずれかを選択し、それを特定のデータ パケットに使用します。

（注） ロード シェアリングでは、特定のフローに含まれるすべてのパケットに対して同じパスが使用されます。フローは、ユーザが設定したロード シェアリング方式によって定義されます。たとえば、送信元/宛先のロード シェアリングを設定すると、送信元 IP アドレスと宛先 IP アドレスのペアが同じであるすべてのパケットが同じパスをたどります。

ユニキャスト FIB のロード シェアリング アルゴリズムを設定するには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **ip load-sharing address** {**destination port destination** | **source-destination** [**port source-destination]** | **source** } [**exclude-l3-proto**] **hardware lb-keyshift** *value* **lb-2nd-heir-keyshift** *value* [**universal-id** *seed*] [**rotate** *rotate*] [**concatenation**]
2. （任意） **show ip load-sharing**
3. （任意） **show routing hash** *source-addr dest-addr* [*source-port dest-port*] [**vrf** *vrf-name*]

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**ip load-sharing address** {**destination port destination** | **source-destination** [**port source-destination]** | **source** } [**exclude-l3-proto**] **hardware lb-keyshift** *value* **lb-2nd-heir-keyshift** *value* [**universal-id** *seed*] [**rotate** *rotate*] [**concatenation**] 例： `ip load-sharing address source-destination port source-destination hardware lb-keyshift 1 lb-2nd-hier-keyshift 10`	データ トラフィックに対するユニキャスト FIB のロード シェアリング アルゴリズムを設定します。 （注） Cisco Nexus 9808/9804 スイッチでは、**ip load-sharing address** 構成時に **address source-destination port source-destination** オプションのみがサポートされます。

	コマンドまたはアクション	目的
		Cisco NX-OS リリース 10.3(3)F 以降では、Cisco Nexus 9600-R/RX ライン カードでのみ、**IHB_ECMP_LB_KEY_CFG** テーブルの次のパラメータをサポートするために **hardware** オプションが追加されています。 • **lb-keyshift**：ロードバランシングの **ECMP_LB_KEY_SHIFT** 値を設定します。指定できる範囲は、1 ～ 10 です。 • **lb-2nd-hier-keyshift**：ロード バランシングの **ECMP_2ND_HIER_LB_KEY_SHIFT** 値を設定します。指定できる範囲は、1 ～ 10 です。 Cisco NX-OS リリース 10.5(1)F 以降、 **exclude-l3-proto** オプションが導入されました。このオプションを使用すると、Cisco Nexus 9300-GX2 シリーズ スイッチでのネクストホップ選択時に、ECMP ハッシュから IP プロトコルを除外できます。 次のオプションは、すべての IP ロード シェアリング設定で使用できます。 • **universal-id** オプションは、ハッシュ アルゴリズムのランダム シードを設定することにより、フローをあるリンクから別のリンクにシフトします。 汎用 ID を設定する必要はありません。ユーザが設定しなかった場合は、Cisco NX-OS が汎用 ID を選択します。*universal-id* の範囲は 1 ～ 4294967295 です。 • **rotate** オプションを使用すると、ハッシュ アルゴリズムは、リンク ピッキングの選択をローテーションさせます。これは、ネットワーク内のすべてのノードが同じリンクを継続的に選択しないようにするためです。これは、ハッシュ アルゴリズムのビット パターンに影響を与えることによって機能します。このオプションは、あるリンクから別のリンクにフローをシフトし、最初の ECMP レベルからすでにロード バランシング（極性化）されているトラフィックのロード バランシングを複数のリンク間で行います。 *rotate* 値を指定すると、64 ビットのストリームが、循環回転でのそのビット位置から解釈され

	コマンドまたはアクション	目的
		ます。*rotate* 値の範囲は 1 ～ 63 で、デフォルトは 32 です。 （注） 多層レイヤ 3 トポロジでは、極性が発生する可能性があります。極性を回避するには、トポロジの各層で異なる循環ビットを使用します。 （注） ポート チャネルの rotation 値を設定するには、**port-channel load-balance src-dst ip-l4port rotate** *rotate* コマンドを使用します。このコマンドの詳細については、『*Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*』を参照してください。 • **concatenation** オプションを使用すると、ECMP のハッシュ タグ値とポート チャネルのハッシュ タグ値がひとつに結合され、より強力な 64 ビットのハッシュを使用できるようになります。このオプションを使用しない場合、ECMP のロードバランシングおよびポート チャネルのロードバランシングを個別に制御できます。デフォルトではディセーブルになっています。
Step 2	（任意） **show ip load-sharing** 例： `switch(config)# show ip load-sharing address source-destination`	データ トラフィックに対するユニキャスト FIB のロード シェアリング アルゴリズムを表示します。
Step 3	（任意） **show routing hash** *source-addr dest-addr* [*source-port dest-port*] [**vrf** *vrf-name*] 例： `switch(config)# show routing hash 192.0.2.1 10.0.0.1`	ユニキャスト RIB とユニキャスト FIB が特定の送信元と宛先アドレス のペアに使用するルートを表示します。送信元アドレスと宛先アドレスの形式は x.x.x.x です。送信元ポートと宛先ポートの範囲は 1 ～ 65535 です。VRF 名には最大 64 文字の英数字文字列を指定します。大文字と小文字は区別されます。

例

次に、送信元/宛先ペアのために選択されたルートを表示する例を示します。

```
switch# show routing hash 10.0.0.5 192.0.0.2
Load-share parameters used for software forwarding:
load-share mode: address source-destination port source-destination
Universal-id seed: 0xe05e2e85
```

```
Hash for VRF "default"
Hashing to path *172.0.0.2 (hash: 0x0e), for route:
```

次に、**show ip load-sharing** コマンドの出力例を示します。

```
switch(config)# show ip load-sharing
IPv4/IPv6 ECMP load sharing:
Universal-id (Random Seed): 1913447906
Load-share mode : address source
Exclude L3 proto from ECMP hashing : Enabled
Rotate: 32
switch(config)#
```

ダイナミック ロード バランシング

ダイナミックロードバランシングについて

ダイナミックロードバランシング（DLB）は、トラフィックを、回送メトリックの点で同じコストを持つ複数のパスまたはリンクに分散するために使用される、ネットワーキング技術です。これはIP レイヤ（OSI モデルのレイヤ3）で実行されるもので、最新のネットワーキングハードウェアの多くに実装されています。これには、Cisco NX-OS リリース 10.5(1)F 以降を搭載する、Nexus 9000 シリーズクラウドスケールスイッチなども含まれます。

ECMP は、ネットワーク内の任意の 2 ポイント間を流れるトラフィックに、並列の複数のパスを許可することで、アプリケーションで使用可能な帯域幅を増やすために使用されます。複数の等コスト パスを持つ接続先にパケットを転送する必要がある場合、ルータはハッシュアルゴリズムを使用して、そのパケットに使用する経路を決定します。このアルゴリズムでは、通常、送信元と宛先の IP アドレス、送信元と宛先のポート番号、場合によってはプロトコルタイプなどのパラメータが考慮されます。

従来のロードバランシングでは、ネットワークトポロジが変更されたり、ネットワーク管理者が手動で再構成したりしない限り、パスが時間の経過とともに変化することはありません。対照的に、レイヤ 3 ECMP のダイナミックロードバランシングでは、ネットワークの現在の状態に応じてパスの選択を変更できるようになっています。ルータまたはスイッチは、各パスのトラフィック負荷をモニターし、リンク使用率が最も低いパスを選択して、使用可能なすべてのパスにトラフィックをより適切に分散できます。したがって、 Nexus 9000 スイッチのレイヤ 3 ECMP DLB を使用すると、ネットワーク内の複数の等コストパスにトラフィックを効率的に分配することができます。

レイヤ 3 ECMP DLB は、特にバックエンド AI/ML トレーニングネットワークで使用される、リーフ/スパイン型アーキテクチャを備えた RDMA over Ethernet（RoCE）でサポートされます。DLB を備えたファブリックは、PFC とともに ECN と組み合わせると、使用率の向上、低遅延、および損失のないファブリックとして最適なネットワーク動作を実現します。

機能

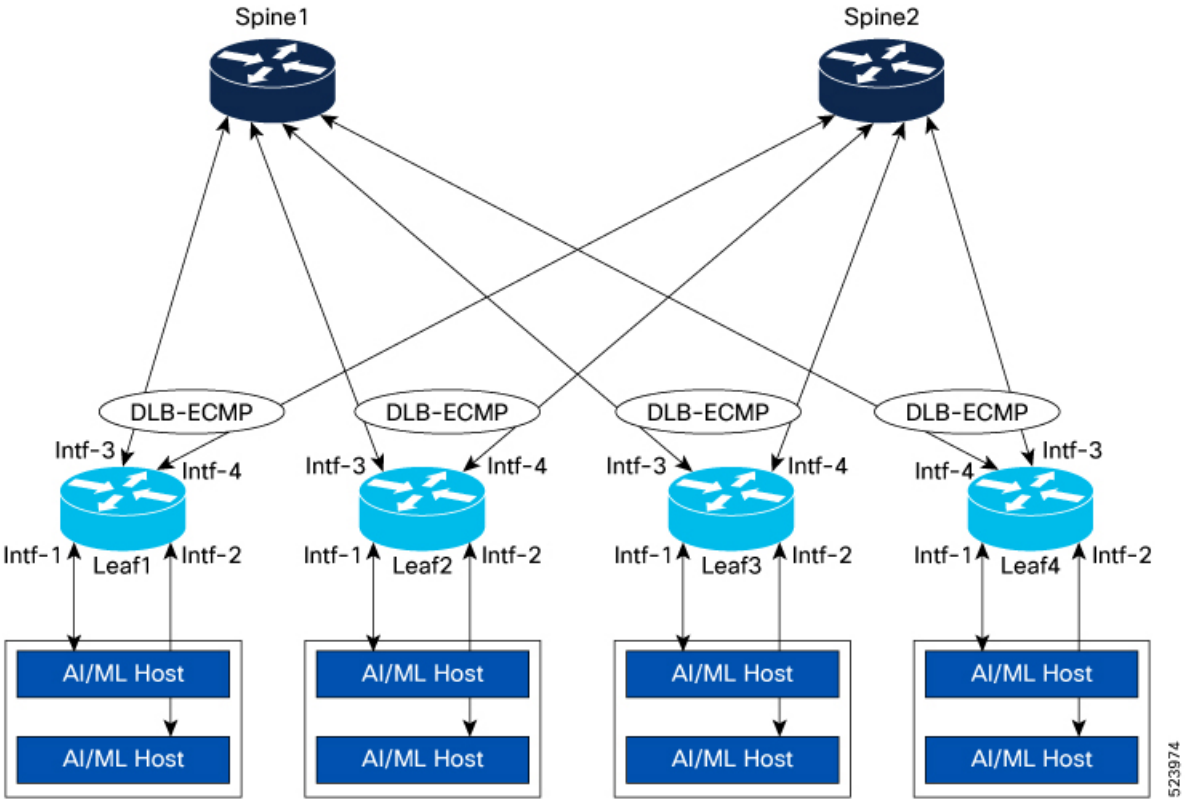
- リンクまたはノードに障害が発生した場合のコンバージェンスの高速化
- 利用可能なネットワーク パスを最大限に使用する
- リンクまたはノードに障害が発生した場合に冗長性を提供する

- すべてのパスにトラフィックを均等に展開させることで、輻輳を最小限に抑える
- インフラストラクチャの追加や特殊なインフラストラクチャを必要とせずに、ネットワーク全体のパフォーマンスを向上させる

トポロジ

DLB トポロジは、人工知能（AI）および機械学習（ML）のトレーニング ネットワークで役立ちます。これらのネットワークは、図に示すスパインリーフアーキテクチャを使用しています。ここでは、AI および ML ホスト（サーバ）は、リーフスイッチのインターフェイス 1（Intf-1）とインターフェイス 2（Intf-2）に接続されています。リーフ スイッチの Intf-3 と Intf-4 は、2 つのスパイン（スパイン 1 とスパイン 2）に接続されています。AI および ML ホスト間でデータ（トレーニングデータなど）を同期している間、トレーニングデータは、スパインリーフファブリックを介してすべてのホスト間で転送されます。

図 *43 : DLB* トポロジ



リーフスイッチは複数のリンクでスパインに接続されているため、 ECMP は複数のリンク間でトラフィックを負荷分散するために使用されます。従来のネットワークとは異なり、 AI および ML トレーニングネットワークでは、（一意の 5 タプル IP フィールドを持つ）トラフィックフローの数が少なくなります。フローが少ないと、従来の ECMP では偏りの問題（冗長パスの使用が最適化されない）が発生し、一部のリンクまたはインターフェイスでオーバーサブスクリプションが発生する可能性があります。少数のリンクのオーバーサブスクリプションは、アーキテクチャに

他にもリンクが存在していても、その使用率が低い、または全く使用されないことにつながり、トラフィック損失を引き起こす可能性があります。

ECMP DLB 機能は、すべてのリンクが適切に使用されるようにすることで、リンクが全く、または少ししか使用されていない状態を解消します。必要に応じて、使用率もカスタマイズできます。ECMP グループの一部であるすべてのポートで DLB が有効になっている場合、新しいフローごとに、使用可能なリンクの中から Tx リンク使用率が最小のリンクが選択されます。上の画像では、Intf-3 と Intf-4 で DLB が有効になっています。intf-3 が完全に使用されていて、新しいフローを受信した場合、intf-4 が選択されます。従来の ECMP では、オーバーサブスクリプションが生じていても、Intf-3 が選択される可能性があります。

ECMP DLB は静的ピン接続もサポートします。これにより、ユーザーは特定の送信元ポートからのトラフィックを常に特定の DLB 対応出力ポートに送信できます。画像では、Intf-3 と Intf-4 がメンバーである DLB ECMP グループを取っているトラフィックに当てはまります。ユーザーは、Intf-1 からのトラフィックが常に Intf-3 を取り、Intf-2 からのトラフィックが常に Intf-4 を取るようにピン留めできます。

ダイナミック ロード バランシングの主要な概念

高速リンクフェールオーバー

Nexus 9000 スイッチのレイヤ 3 ECMP ロード バランシングの背景における高速リンク フェールオーバーとは、ネットワークが物理リンク障害に迅速に対応し、回復できるようにする機能のことです。ECMP グループで使用されているリンクに障害が発生すると、高速リンク フェールオーバーは、従来のルーティングプロトコルのコンバージェンス時間を待たずに、トラフィックを残りの動作中のリンクにすぐにリダイレクトします。

Nexus 9000 スイッチで実行されているレイヤ 3 ECMP DLB 機能は、リンク フェールオーバーをハードウェアによって検出し、残りのリンクから新しいリンクを自動的に選択します。これはハードウェア層で行われるため、コンバージェンスが高速化されます。

Dynamic Rate Estimator

Dynamic Rate Estimator（DRE）は、現在のリンク使用率を測定するためにハードウェアに実装されています。DLB における DRE の役割は、さまざまなリンクのトラフィックレートをリアルタイムで推定することです。このリアルタイム分析により、スイッチはトラフィックを分散する際に、より多くの情報に基づいた決定を行うことができ、単一の経路が過飽和にならないようにします。新しいフローを開始するとき、DLB は DRE メトリックを使用して、DLB ECMP グループ内の複数の経路内で使用率が最も低いパスを決定します。

任意の時点で、DLB 対応インターフェイスは、リンクの使用率と構成された DRE しきい値に基づいて、レベル 1 からレベル 7 の DRE レベルのいずれかになります。レベル 1 は最も低い使用率を示し、レベル 7 は最も高い使用率を示します。DLB は判断する際、常に最も低い DRE レベルのリンクを選択します。複数のリンクが、同じ最も低いDRE レベルである場合、それらの中のリンクの 1 つがランダムに選択されます。詳細については、「リンク使用率レベルの計算」を参照してください。

モード

レイヤ 3 ECMP ダイナミック ロード バランシングは、グローバル構成で次のいずれかのモードをサポートします。

- フローレット ロードバランシング（FLB）：このモードでは、DRE メトリックに基づいてフローレット レベルでロードバランシングを実行します。これは、デフォルトのモードです。
- パケット単位のロードバランシング（PLB）：このモードでは、フローレットレベルではなくパケット単位でロードバランシングの判断が行われます。

FLB

フローレットは、フローからのパケットのバーストであり、5 タプル（つまりパケットから選択されたフィールド）で識別されます。並べ替えを発生させることなく個別にルーティングできるように、十分に大きなギャップで区切られています。

フローレットは、DLB がフローレットモードで動作するときに使用されるトラフィックの単位です。フローレットごとに、ポート単位の DRE によって示される、Tx 使用率が最小のものが、最適な発信ポートとしてハードウェア によって選択されます。使用率がすべてのポートで同じ場合、ポートの 1 つがランダムに選択されます。

あるフローレット用にポートが選択されると、そのフローからの後続のすべてのパケットに同じポートが使用されます。新しいポートは、構成されたフローレットエージング時間を超えるパケット間ギャップがフローレットにある場合、または現在使用されているポートがダウンした場合にのみ、選択されます。

PLB

パケット単位のロードバランシング（PLB）は、エンドポイント（スマート NIC など）でパケットの並べ替えが可能なシナリオで使用できます。このモードは、DLB ECMP 内の使用可能なリンク全体にトラフィックを分散させることにより、ネットワークの輻輳を軽減します。フロー内のパケットごとに、新しい出力ポートが選択されます。そのため、同じフローからのパケットが複数のパスを介して送信され、パケットの順序が変更される可能性があります。ポート選択プロセスでは DRE が使用されます。つまり、どのパケットでも、DRE メトリックが最小のポートが選択されます。DRE がすべてのポートで同じである場合、ポートの 1 つがランダムに選択されます。

静的ピン接続

静的ピン接続は、DLB でサポートされています。静的ピン接続では、送信元ポートは、DLB 対応の ECMP グループの一部である宛先ポートにピン止めされます。このポートがこのフローに使用される DLB ECMP グループの一部である場合、この送信元ポートからのすべてのトラフィックは、ピン止めされた宛先ポートに送信されます。前面パネルのポート（ブレークアウトポートを含む）は、静的ピン接続の送信元インターフェイスとして使用できます。宛先インターフェイスは、DLB インターフェイスリストに含まれている必要があります。

静的ピン接続が有効になっている場合、静的ピン接続は DLB DRE ベースのポート選択を上書きします。

DLB ポートが静的ピン接続の宛先ポートとして使用されている場合、そのポートの静的ピン接続構成が削除されない限り、このポートをDLBインターフェイスリストから削除することはできません。

（注） 静的ピン接続またはPLB モードについては、どちらか一方のみを有効にできます。両方を有効にすることはできません。

（注） Cisco NX-OS リリース 10.5(2)F 以降、システムは最大 512 の静的ピン接続ペアをサポートします。

ダイナミックロードバランシングに関するガイドラインと制限事項

レイヤ3ダイナミックロードバランシングのガイドラインと制限事項は、次のように分類されて、それぞれの項に記載されています。

- ECMP グループ
- 機能のサポート
- ポート
- DLB パラメータ

ECMP グループ

- DLB を有効にするかどうかの判断は、次の 3 つの条件に基づき、ECMP グループの作成時に行う必要があります。
  - ECMP グループのすべてのメンバーが、DLB 対応インターフェイスリストに含まれています。ECMP グループに DLB インターフェイスリストにないメンバーが 1 つ以上ある場合、その ECMP グループには通常の ECMP を使用します。
  - ECMP グループのメンバーは、L3 インターフェイスのみです。ブレークアウトポート、サブインターフェイス、 SVI、およびポートチャネルを DLB ECMP グループのメンバーにすることはできません。
  - ECMP は重み付け ECMP またはレジリエント ECMP ではありません。
- レジリエント ECMP機能と DLB 機能を同時に有効にすることはできません。レジリエント ECMP の詳細については、Cisco Nexus 9000 シリーズ NX-OS インターフェイス構成ガイドを参照してください。
- 重み付け ECMP グループの場合、DLB は適用されません。重み付け ECMP の詳細については、Cisco Nexus 9000 シリーズ NX-OS ユニキャスト ルーティング構成ガイドを参照してください。

- show routing hash コマンドは、DLB ECMP グループを使用するルートでは機能しません。これは、DLB が有効になっている場合、ポートの選択は静的ハッシュではなくリンク使用率に基づいて動的に行われるためです。
- DLB ECMP スケールに達した場合、または何らかの条件により DLB を有効にできない場合は、DLB なしの通常の ECMP が使用されます。

（注） DLB 対応 ECMP グループのメンバーポートの 1 つがダウンすると、ハードウェアによって、トラフィックを送信するためのポートがすぐに選択されます。これにより、リンク障害状態でもトラフィック損失が無視できるようになります。

機能のサポート

- この機能は、以下でのみサポートされています。
  - レイヤ 3 物理インターフェイス
  - IP ルーテッドファブリックおよび VXLAN ファブリック
  - 9300-FX3、GX、GX2、および HX TOR プラットフォーム。
- この機能は、ラインカード拡張モジュール（LEM）および N9K-C9408 を搭載した TOR ではサポートされません。
- 出力インターフェイスで構成された出力アクセスリストポリシー、出力 QOS ポリシー、および TX SPAN は、 ECMP DLB を使用するフローには適用されません。
- MPLS/GRE トンネルは DLB ECMP を使用せず、通常の ECMP にフォールバックします。
- この機能を使用する場合、特に使用される DLB ECMP スケールが高い場合は、system pic-core オプションを使用することを推奨します。
- DLB は、ポリシーベースルーティング（PBR）ロジックが適用されるトラフィックフローには適用されません。これらのフローは、通常の ECMP 機能を使用します。
- MTU は、DLB フローで使用されるパケットの最大サイズに基づいて、すべての DLB 対応インターフェイスで構成する必要があります。そうしなかった場合、トラフィックは出力ドロップとして出力インターフェイスでドロップされます。
- ユニキャスト IPv4 および IPv6 トラフィックのみがサポートされています。

ポート

- ブレークアウトポート、ポートチャネル、SVI、ポートチャネルメンバー、またはサブインターフェイスを DLB 対応インターフェイスリストに含めることはできません。
- 最大 63 個の物理ポートを DLB インターフェイスリストに含めることができます。

DLB パラメータ

このセクションでは、 MAC、エージング、DRE しきい値、モード、静的ピン接続などの DLB 関連パラメータのガイドラインを示します。

- DLB インターフェイスリストを初めて構成する場合、または変更した場合には、スイッチをリロードして構成を有効にする必要があります。
- フローレットエージングタイムは、ファブリックでのラウンドトリップ時間に基づいて選択します。そうしなかった場合、フローの順序が変わる可能性があります。
- DLB 構成のすべての DLB 関連パラメータは、有効な適用済み DLB インターフェイスリストがある場合にのみ、ハードウェアでプログラムされます。
- MAC、エージング、モード、または DRE しきい値構成のいずれかを削除すると、すべてのパラメータがデフォルト値に設定されます。
- ポートをDLBインターフェイスリストに追加した後、ポートがブレークアウトポートに変更されるか、POの一部になるように追加されるか、このインターフェイスでサブインターフェイスが作成されるかすると、DLB は、ポートを含む ECMP グループに対して有効ではなくなります。ユーザーは、DLB インターフェイスリストからポートを削除する必要があります。
- DRE しきい値を変更すると、DLB 対応フローのトラフィックに一時的な影響が及ぶ可能性があります。これは中断を伴うトリガーです。
- DLB MAC 構成は、ファブリック内のすべてのノードで同一である必要があります。これらのフローを受信するファブリック内の接続ノードで DLB MAC 構成を変更せずに、スイッチでその構成を変更すると、トラフィックはドロップされます。
- 静的ピン接続モードと、パケットごとの DLB モードは、同時にサポートできません。
- ブレークアウトポートは、静的ピン接続の送信元インターフェイスの一部にすることができます。ただし、サブインターフェイスとポートチャネルを、静的ピン接続の送信元インターフェイスの一部にすることはできません。

ダイナミック ロード バランシングの構成

レイヤ 3 のダイナミック ロード バランシングを構成するには、**hardware profile dlb** サブモードで次のコマンドを実行します。

手順の概要

1. **configure terminal**
2. **hardware profile dlb**
3. **dlb-interface** <*interface_range*>
4. （任意） **dre-thresholds** [**level-1** *percentage_1* | **level-2** *percentage_2* | **level-3** *percentage_3* | **level-4** *percentage_4* | **level-5** *percentage_5* | **level-6** *percentage_6* | **level-7** *percentage_7*]
5. （任意） **flowlet-aging** *usec*
6. （任意） **mac-address** *macaddr*
7. （任意） **mode** [**flowlet** | **per-packet**]

8. （任意） **static-pinning**
9. （任意） **source** 送信元物理インターフェイス **destination** 接続先物理インターフェイス

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例： `switch# configure terminal`	グローバル コンフィギュレーション モードを開始します。
Step 2	**hardware profile dlb** 例： `switch(config)# hardware profile dlb`	ハードウェア プロファイルのダイナミック ロードバランシングモードを開始します。
Step 3	**dlb-interface** <*interface_range*> 例： `switch(config-dlb)# dlb-interface Eth1/5,Eth1/7,Eth1/17,Eth1/21,Eth1/26`	DLB を有効にするインターフェイスのリストを指定します。このリストは動的には変更できません。インターフェイスリストを有効にするには、スイッチのリロードが必要です。カンマ区切りで、インターフェイスを追加します。 （注） • インターフェイスリストを変更した場合は、構成を有効にするためにリロードが必要です。 • 現在の適用済みリストを確認するには、**show hardware profile dlb** コマンドを実行します。 • インターフェイスリストへの部分的な追加や削除はサポートされていません。新しいインターフェイスリストが指定されると、構成は、新しいリストで置き換えられます。
Step 4	（任意） **dre-thresholds** [**level-1** *percentage_1* | **level-2** *percentage_2* | **level-3** *percentage_3* | **level-4** *percentage_4* | **level-5** *percentage_5* | **level-6** *percentage_6* | **level-7** *percentage_7*] 例： `switch(config-dlb)# dre-thresholds level-1 15 level-2 20 level-3 30 level-4 15 level-5 10 level-6 5 level-7 5`	レベル 1 からレベル 7 までの DRE レベルを定義します。レベルごとに構成される値は、前のレベルからのポート帯域幅の使用率の範囲です。指定するすべてのレベルの合計は、100 にする必要があります。DRE しきい値レベルを構成しなかった場合は、デフォルト値として、30、20、15、10、10、10、および 5 が使用されます。 DRE レベルのリンク使用率の詳細については、リンク使用率レベルの計算を参照してください。

	コマンドまたはアクション	目的
Step 5	（任意） **flowlet-aging** *usec* 例： `switch(config-dlb)# flowlet-aging 600`	フローレットのエージング時間を設定します。値はマイクロ秒単位です。デフォルトは 500 マイクロ秒で、最大値は 2 秒つまり 2000000 マイクロ秒です。 （注） フローレットのエージング時間は慎重に選択してください。不用意に選択すると、フローの順序が変わってしまう可能性があります。
Step 6	（任意） **mac-address** *macaddr* 例： `switch(config-dlb)# mac-address aa:bb:cc:dd:ee:ff`	DLB MAC アドレスを構成します。このアドレスは、DLB を使用するすべてのフローのネクストホップ MAC アドレスとして使用されます。この DLB MAC は、出力インターフェイスの学習されたネクストホップMACアドレスを変更するものとして、DLB フローの宛先 MAC を書き換えるために使用されます。 次の注意事項および制約事項が適用されます。 • このコマンドを構成しなかった場合、機能の初期化中に使用されるデフォルトの DLB MAC アドレス（00:CC:CC:CC:CC:CC）が、デフォルトの DMAC として使用されます。 • DLB のMAC アドレスを構成すると、デフォルトの MAC は、新しい構成可能 MAC アドレスで置き換えられます。 • このスイッチで、この DLB MAC を宛先 MAC として受信されたすべてのパケットは、ルーテッド パケットとして扱われます。 • この設定を適用する場合は、ファブリック内の他のすべてのノードが同じ DLB MAC で構成されていることを確認してください。 • dlb-interface リストが適用されていない場合、DLB MAC を追加のルータ MACとして使用することはできません。 • ブロードキャストおよびマルチキャスト MAC アドレスを DLB MAC アドレスとして設定することはできません。
Step 7	（任意） **mode** [**flowlet** | **per-packet**] 例： `switch(config-dlb)# mode flowlet`	フローレットまたはパケット単位の DLB モードを有効にします。デフォルトのモードはフローレットです。 （注）

	コマンドまたはアクション	目的
		パケット単位モードの場合、静的ピン接続を有効にすることはできません。
Step 8	（任意） **static-pinning** 例： `switch(config-dlb)# static-pinning`	静的ピン接続機能を構成します。 （注） 静的ピン接続の場合、パケット単位モードを有効にすることはできません。
Step 9	（任意） **source** 送信元物理インターフェイス **destination** 接続先物理インターフェイス 例： `switch(config-dlb-static-pinning)# source ethernet 1/1 destination ethernet 1/2`	静的ピン接続の送信元インターフェイスと接続先インターフェイスを構成します。ただし、これらは物理インターフェイスのみでなければなりません。つまり、前面パネルのイーサネットインターフェイスです。SVI、ポートチャネル、またはサブインターフェイスを送信元または接続先インターフェイスにすることはできません。 次の注意事項および制約事項が適用されます。 • 接続先インターフェイスは、DLB が適用または構成されたインターフェイスリストの一部である必要がありますが、送信元インターフェイスをこのリストの一部にすることはできません。 • Cisco NX-OS リリース 10.5(2)F 以降では、DLB が適用または構成された interface-list のインターフェイスは、静的ピン接続の送信元インターフェイスとしても使用できます。 • 2 つの構成で同じ送信元インターフェイスが使用されている場合、最初の接続先インターフェイスは 2 番目の接続先インターフェイスで置き換えられます。送信元インターフェイスは同じだからです。 • ブレークアウトポートは、送信元インターフェイスとして構成できます。 • ポートでブレークアウトまたはブレークアウト操作が実行されていない場合、ユーザーは DLB または静的ピン接続構成を更新する必要があります。 • 静的ピン接続の接続先インターフェイスとして構成されているインターフェイスは、DLB インターフェイスリストから削除できません。削除するには、最初に静的ピン接続構成を削除して、

	コマンドまたはアクション	目的
		その後 DLB インターフェイスリストからインターフェイスを削除します。

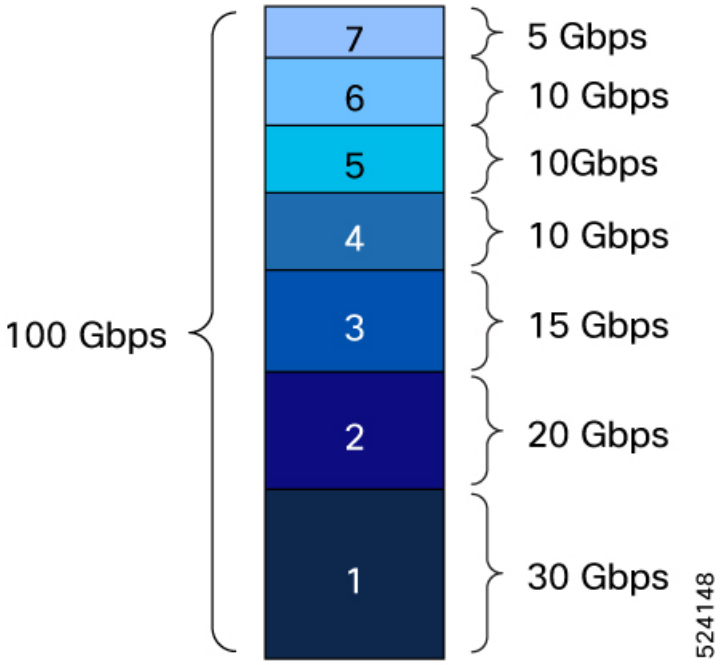
リンク使用率レベルの計算

各レベルのリンク使用率は、次のように計算されます。

- 現在のレベル範囲の開始：以前のすべてのレベルに指定された % 値の合計。
- 現在のレベル範囲の終了：現在のレベル範囲の開始 + 現在のレベルの値。

次に例を示します。

Level-1：30、Level-2：20、Level-3：15、Level-4：10、Level-5：10、Level-6：10、Level-7：5。



- レベル 5 範囲の開始：75%（30 + 20 + 15 + 10）
- レベル 5 範囲の終了：85%（75 + 10）

DRE しきい値レベルのガイドライン

Dynamic Rate Estimator （DRE）のしきい値レベルに関連するガイドラインと制限事項を次に示します。

- 上記のロジックは、いずれかのレベルがゼロ（0）のシナリオに適用されます。レベルを追加すると、そのレベルは、ゼロ以外の値を持つ前のレベルと同じ範囲になります。

  次の例を参考にしてください。

Level-1：30、Level-2：0、Level-3：35、Level-4：10、Level-5：10、Level-6：10、Level-7：5。

- レベル 2 の範囲：レベル 2 はレベル 1 と同じ、つまり 30% になります。

- 以前の値がすべてゼロの場合、現在のレベルは最初に指定されたゼロ以外のレベルになります。

  次の例を参考にしてください。

  Level-1：0、Level-2：0、Level-3：0、Level-4：0、Level-5：50、Level-6：30、Level-7：20。

  - この場合、リンク使用率の開始レベルは Level-5 になります。

- すべてのレベルを指定してください。いくつかのレベルが指定されていない場合、それらはゼロと見なされます。

  次の例を参考にしてください。

  Level-1：50、Level-2：0、Level-3：0、Level-4：0、Level-5：0、Level-6：30、Level-7：20

  - レベル 2、3、4、および 5 はゼロと見なされます。

ダイナミックロードバランシングの構成例

```
switch# configure terminal
switch(config)# hardware profile dlb
switch(config-dlb)# dlb-interface Eth1/5,Eth1/7,Eth1/17,Eth1/21,Eth1/26 switch(config-dlb)#
 dre-thresholds level-1 15 level-2 20 level-3 30 level-4 15 level-5 10 level-6 5 level-7 5
 switch(config-dlb)# flowlet-aging 600 switch(config-dlb)# mac-address aa:bb:cc:dd:ee:ff
switch(config-dlb)# mode flowlet switch(config-dlb)# static-pinning
switch(config-dlb-static-pinning)# source ethernet 1/1 destination ethernet ½
```

ダイナミックロードバランシングの確認

ダイナミックロードバランシングの構成情報を表示するには、次のいずれかのコマンドを実行します。

コマンド	目的
show hardware profile dlb	DLB の構成を表示します。 （注） • [構成済みインターフェイスリスト（Configured Interface-list）]：コマンド ラインインターフェイスを使用して構成されている現在のインターフェイスのリストを提供します。リロード後、同じリストが、適用済みインターフェイスリストに入力されます。 • [適用済みインターフェイスリスト（Applied Interface-list）]：DLB に現在使用されているインターフェイスのリストを提供します。
show system config reload-pending	リロード前の、保留されている構成を表示します。DLB で、インターフェイスリストの構成に変更があった場合、適用が保留されているインターフェイスリストが表示されます。

（注） **show routing hash** コマンドは、DLB ECMP グループを使用するルートでは機能しません。

表示コマンドの出力

show hardware profile dlb コマンドの出力例を次に示します。

```
switch# show hardware profile dlb
DLB Configurations:
===================


Enabled:                yes
Mode:                   flowlet
Mac-address:            aa:bb:cc:dd:ee:ff
Flowlet aging time:     600 usec(s)
DRE-thresholds:
        Level-1:15
        Level-2:20
        Level-3:30
        Level-4:15
        Level-5:10
        Level-6:5
        Level-7:5
DLB interface list:
--------------------

Configured Interface-list (size: 5):
        Eth1/5,Eth1/7,Eth1/17,Eth1/21,Eth1/26
```

```
Applied interface-list (size: 5):
        Eth1/5,Eth1/7,Eth1/17,Eth1/21,Eth1/26

Static-pinning enabled: yes

DLB static-pinning pairs:
---------------------------

static-pinning pairs (1):

        source: Eth1/1    dest: Eth1/5
```

show system config reload-pending コマンドの出力例を次に示します。

```
switch# show system config reload-pending

Following config commands require copy r s + reload :
=====================================================
0       hardware profile dlb ; dlb-interface Eth1/5,Eth1/7,Eth1/17,Eth1/21,Eth1/26
=====================================================
```

ダイナミックロードバランシングのトラブルシュート

整合性チェッカーは、次のように、DLB ECMP を使用するルートをトラブルシュートするために使用できます。

- グローバル整合性チェッカー
  - **test consistency-checker forwarding ipv4 unicast**
  - **show consistency-checker forwarding ipv4 unicast**

  サンプル出力

  ```
  Leaf1# test consistency-checker forwarding ipv4 unicast
  Consistency check started.
  Leaf1#
  Leaf1#
  Leaf1# show consistency-checker forwarding ipv4 unicast
  IPV4 Consistency check : table_id(0x1)
  Execution time : 28 ms ()
  No inconsistent adjacencies.
  No inconsistent routes.
  Consistency-Checker: PASS for ALL
  ```

- シングルルート整合性チェッカー
  - **show consistency-checker forwarding single-route ipv4** *ipv4 address* **vrf** *vrf*

  サンプル出力

  ```
  Leaf1# show consistency-checker forwarding single-route ipv4 64.60.60.0/24 vrf default


  Consistency checker passed for 64.60.60.0/24
  Leaf1#
  ```

ルーティング情報と隣接情報の表示

ルーティング情報と隣接情報を表示するには、任意のモードで次のコマンドを使用します。

コマンド	目的
show {**ip** | **ipv6**} **route** [*route-type* | **interface** *interface-type number* | **next-hop**] `switch# show ip route`	ユニキャスト ルート テーブルを表示します。*route-type* 引数には、1 つのルート プレフィックス、ダイレクト、スタティック、またはダイナミック ルーティング プロトコルを指定できます。**?** コマンドを使用すると、サポートされているインターフェイスが表示されます。
show {**ip** | **ipv6**} **adjacency** [*prefix* | *interface-type number* [**summary**] | **non-best**] [**detail**] [**vrf** *vrf-id*] 例： `switch# show ip adjacency`	隣接関係テーブルを表示します。引数の範囲は次のとおりです。 • *prefix*：任意の IPv4、または IPv6 プレフィックス アドレス。 • *interface-type number*：**?** コマンドを使用して、サポートされるインターフェイスを表示します。 • *vrf-id*：最大 64 文字の英数字文字列を指定します。大文字と小文字は区別されます。
show {**ip** | **ipv6**} **routing** [*route-type* | **interface** *interface-type number* | **next-hop** | **recursive-next-hop** | **summary** | **updated** {**since** | **until**} *time*] 例： `switch# show routing summary`	ユニキャスト ルート テーブルを表示します。*route-type* 引数には、1 つのルート プレフィックス、ダイレクト、スタティック、またはダイナミック ルーティング プロトコルを指定できます。**?** コマンドを使用すると、サポートされているインターフェイスが表示されます。

次に、ユニキャスト ルート テーブルを表示する例を示します。

```
switch# show ip route
IP Route Table for Context "default"
'*' denotes best ucast next-hop '**' denotes best mcast next-hop
```

```
'[x/y]' denotes [preference/metric]

0.0.0.0/0, 1 ucast next-hops, 0 mcast next-hops
   *via 10.1.1.1, mgmt0, [1/0], 5d21h, static
0.0.0.0/32, 1 ucast next-hops, 0 mcast next-hops
   *via Null0, [220/0], 1w6d, local, discard
10.1.0.0/22, 1 ucast next-hops, 0 mcast next-hops, attached
   *via 10.1.1.55, mgmt0, [0/0], 5d21h, direct
10.1.0.0/32, 1 ucast next-hops, 0 mcast next-hops, attached
   *via 10.1.0.0, Null0, [0/0], 5d21h, local
10.1.1.1/32, 1 ucast next-hops, 0 mcast next-hops, attached
   *via 10.1.1.1, mgmt0, [2/0], 5d16h, am
10.1.1.55/32, 1 ucast next-hops, 0 mcast next-hops, attached
   *via 10.1.1.55, mgmt0, [0/0], 5d21h, local
10.1.1.253/32, 1 ucast next-hops, 0 mcast next-hops, attached
   *via 10.1.1.253, mgmt0, [2/0], 5d20h, am
10.1.3.255/32, 1 ucast next-hops, 0 mcast next-hops, attached
   *via 10.1.3.255, mgmt0, [0/0], 5d21h, local
255.255.255.255/32, 1 ucast next-hops, 0 mcast next-hops
   *via Eth Inband Port, [0/0], 1w6d, local
```

次に、隣接関係情報を表示する例を示します。

```
switch# show ip adjacency
IP Adjacency Table for context 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
```

レイヤ 3 整合性チェッカーのトリガー

レイヤ 3 整合性チェッカーを手動でトリガーできます。

レイヤ 3 整合性チェッカーを手動でトリガーにするには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **test forwarding** [**ipv4** | **ipv6**] [**unicast**] **inconsistency** [**vrf** *vrf-name*] [**module** {*slot* | **all**}]
2. **test forwarding** [**ipv4** | **ipv6**] [**unicast**] **inconsistency** [**vrf** *vrf-name*] [**module** {*slot* | **all**}] **stop**
3. **show forwarding** [**ipv4** | **ipv6**] [**unicast**] **inconsistency** [**vrf***vrf-name*] [**module** {*slot* | **all**}]
4. **show consistency-checker forwarding unicast**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**test forwarding** [**ipv4** | **ipv6**] [**unicast**] **inconsistency** [**vrf** *vrf-name*] [**module** {*slot* | **all**}] 例：	レイヤ 3 整合性チェックを開始します。*vrf-name* には最大 64 文字の英数字文字列を指定します。大文字

	コマンドまたはアクション	目的
	`switch(config)# test forwarding inconsistency`	と-小文字は区別されます。*slot* の範囲は 1 ～ 26 です。
Step 2	**test forwarding** [**ipv4** | **ipv6**] [**unicast**] **inconsistency** [**vrf** *vrf-name*] [**module** {*slot* | **all**}] **stop** 例： `switch(config)# test forwarding inconsistency stop`	レイヤ 3 整合性チェックを停止します。*vrf-name* には最大 64 文字の英数字文字列を指定します。大文字と小文字は区別されます。*slot* の範囲は 1 ～ 26 です。
Step 3	**show forwarding** [**ipv4** | **ipv6**] [**unicast**] **inconsistency** [**vrf***vrf-name*] [**module** {*slot* | **all**}] 例： `switch(config)# show forwarding inconsistency`	レイヤ 3 整合性チェックの結果を表示します。*vrf-name* には最大 64 文字の英数字文字列を指定します。大文字と-小文字は区別されます。*slot* の範囲は 1 ～ 26 です。
Step 4	**show consistency-checker forwarding unicast** 例： `switch(config)# show consistency-checker forwarding unicast`	ユニキャスト ルータのレイヤ 3 整合性チェックの結果を表示します。

FIB 内の転送情報の消去

FIB 内の 1 つまたは複数のエントリを消去できます。FIB のエントリを消去しても、ユニキャスト RIB に影響はありません。

⚠

注意 **clear forwarding** コマンドを実行すると、デバイス上の転送が中断されます。

FIB 内のエントリ（レイヤ 3 の不整合を含む）を消去するには、任意のモードで次のコマンドを使用します。

コマンド	目的
clear forwarding{**ipv4** | **ipv6**} **route** {***** | *prefix*} [**vrf** *vrf-name*] **module** {*slot* | **all**} 例： `switch# clear forwarding ipv4 route * module 1`	FIB から 1 つまたは複数のエントリを消去します。ルートのオプションは次のとおりです。 • *****：すべてのルート。 • *prefix*：任意の IP または IPv6 プレフィックス *vrf-name* には最大 64 文字の英数字文字列を指定します。大文字と-小文字は区別されます。*slot* の範囲は 1 ～ 26 です。

ユニキャスト RIB の最大ルート数の設定

ルーティング テーブルで許可されている最大ルート数を設定できます。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **address-family {ipv4 | ipv6} unicast**
4. **maximum routes** *max-routes* [*threshold* [**reinstall** *threshold*] | **warning -only**]
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context management2`
	`switch(config-vrf)#`
		VRF を作成し、VRF設定モードを開始します。

Step 3	**address-family {ipv4 | ipv6} unicast**
	例：
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af-ipv4)`
		アドレス ファミリ コンフィギュレーション モードを開始します。

Step 4	**maximum routes** *max-routes* [*threshold* [**reinstall** *threshold*] | **warning -only**] 例： `switch(config-vrf-af-ipv4)# maximum routes 300000`	ルーティング テーブルで許可される最大ルート数を設定します。範囲は 1 ～ 4294967295 です。 次の項目を任意で指定できます。 • *threshold*：警告メッセージをトリガーする最大ルート数のパーセンテージ。範囲は 1 ～ 100 です。 • **warning-only**—ルートの最大数を超えた場合に警告メッセージを記録します。 • **reinstall** *threshold*：最大ルート数の上限を超過したために拒否された以前のルートを再インストー

	コマンドまたはアクション	目的
		ルし、それらを再インストールするしきい値を指定します。しきい値の範囲は 1 ～ 100 です。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config-vrf-af-ipv4)# copy running-config startup-config`	この設定変更を保存します。

ルートのメモリ要件の見積もり

一連のルートおよびネクストホップ アドレスが使用するメモリを見積もることができます。

ルートのメモリ要件を見積もるには、任意のモードで次のコマンドを使用します。

コマンド	目的
show routing {**ipv6**} **memory estimate routes** *num-routes* **next-hops** *num-nexthops* 例： `switch# show routing memory estimate routes 5000 next-hops 2`	ルートのメモリ要件を表示します。 *num-routes* の範囲は 1000 ～ 1000000 です。 *num-nexthops* の範囲は 1 ～ 16 です。

ユニキャスト RIB 内のルートの消去

ユニキャスト RIB から 1 つまたは複数のルートを消去できます。

⚠

注意　* キーワードは、ルーティングに深刻な悪影響をもたらします。

ユニキャスト RIB 内の 1 つ以上のエントリを消去するには、任意のコンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
clear {**ip** | **ip4** | **ipv6**} **route** {***** | {*route* | *prefix/length*} [*next-hop interface*]} [**vrf** *vrf-name*] 例： `switch(config)# clear ip route 10.2.2.2`	ユニキャスト RIB とすべてのモジュール FIB から 1 つまたは複数のルートを消去します。ルートのオプションは次のとおりです。 • *****：すべてのルート。 • *route*：個々の IP または IPv6 ルート。 • *prefix*/*length*：任意の IP または IPv6 プレフィックス • *next-hop*：ネクストホップ アドレス。 • *interface*：ネクストホップ アドレスに到達するためのインターフェイス *vrf-name* には最大 64 文字の英数字文字列を指定します。大文字と小文字は区別されます。
clear routing [**multicast** | **unicast**] [**ip** | **ip4** | **ipv6**] {***** | {*route* | *prefix/length*} [*next-hop interface*]} [**vrf** *vrf-name*] 例： `switch(config)# clear routing ip 10.2.2.2`	ユニキャスト RIB から 1 つまたは複数のルートを消去します。ルートのオプションは次のとおりです。 • *****：すべてのルート。 • *route*：個々の IP または IPv6 ルート。 • *prefix*/*length*：任意の IP または IPv6 プレフィックス • *next-hop*：ネクストホップ アドレス。 • *interface*：ネクストホップ アドレスに到達するためのインターフェイス *vrf-name* には最大 64 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ユニキャスト RIB および FIB の確認

ユニキャスト RIB および FIB の設定情報を表示するには、次のいずれかの作業を行います。

コマンド	目的
show forwarding adjacency	モジュールの隣接関係テーブルを表示します。
show forwarding distribution {**clients** | **fib-state**}	FIB の分散情報を表示します。

コマンド	目的
show forwarding interfaces module *slot*	モジュールの FIB 情報を表示します。
show forwarding {**ip** | **ipv4** | **ipv6**} **route**	FIB 内のルートを表示します。
show {**ip** | **ipv6**} **adjacency**	隣接関係テーブルを表示します。
show {**ip** | **ipv6**} **route**	ユニキャスト RIB から受け取った の IPv4 または IPv6 ルートを表示します。
show routing	ユニキャスト RIB から受け取ったルートを表示します。
`show system internal access-list dest-miss stats`	宛先の FIB ルートがないためにドロップされたパケットの統計情報を表示します。DEST MISS とも呼ばれます。出力には、DEST MISS カウンタの増分が表示されます。 （注） Cisco NX-OS リリース 10.1(1) 以降、この機能は Cisco Nexus 9300-FX3 プラットフォーム スイッチでサポートされます。

その他の参考資料

ユニキャスト RIB および FIB の管理に関連する詳細情報については、次の項を参照してください。

- 関連資料

関連資料

関連項目	マニュアル タイトル
EEM の設定	『Cisco Nexus 9000 シリーズ NX-OS システム管理コンフィギュレーション ガイド』

第 17 章

Route Policy Manager の設定

この章は、次の項で構成されています。

- Route Policy Manager について （587 ページ）
- Route Policy Manager の注意事項と制約事項 （598 ページ）
- Route Policy Manager パラメータのデフォルト設定 （599 ページ）
- Route Policy Manager の設定 （600 ページ）
- ルート マップの削除をブロックするグローバル コマンド （621 ページ）
- Route Policy Manager の設定の確認 （621 ページ）
- Route Policy Manager の設定例 （622 ページ）
- 関連項目 （622 ページ）

Route Policy Manager について

Route Policy Manager は、ルート マップおよび IP プレフィックス リストをサポートしています。この機能は、ルート再配布に使用されます。プレフィックス リストには、1 つまたは複数の IPv4 または IPv6 ネットワーク プレフィックスおよび関連付けられたプレフィックス長の値を指定します。プレフィックス リストは、ボーダー ゲートウェイ プロトコル（BGP）テンプレート、ルート フィルタリング、またはルーティング ドメイン間で交換されるルートの再配布などの機能で、単独で使用できます。

ルート マップは、ルートおよび IP パケットの両方に適用できます。ルート フィルタリングおよび再配布は、ルート マップを使用してルートを渡します。

プレフィックス リスト

プレフィックス リストを使用すると、アドレスまたはアドレス範囲を許可または拒否することができます。プレフィックスリストによるフィルタリングでは、ルートまたはパケットのプレフィックスと、プレフィックス リストに指定されているプレフィックスの照合が行われます。特定のプレフィックスがプレフィックス リストのどのエントリとも一致しなかった場合、実質的に拒否されたものと見なされます。

プレフィックス リストに複数のエントリを設定し、エントリと一致したプレフィックスを許可または拒否できます。各エントリにはシーケンス番号が関連付けられています。この番号はユーザが設定できます。シーケンス番号が設定されていない場合は、Cisco NX-OS によって自動的にシーケンス番号が設定されます。Cisco NX-OS はシーケンス番号が最も小さいエントリから順番にプレフィックス リストを評価します。Cisco NX-OS は、所定のプレフィックスと最初に一致したエントリを処理します。一致すると、Cisco NX-OS は permit 文または deny 文を処理し、プレフィックス リストの残りのエントリは評価しません。

（注） プレフィックス リストが空の場合は、すべてのルートが許可されます。

MAC リスト

MAC リストを使用すると、MAC アドレスまたはアドレス範囲を許可または拒否できます。MAC リストは MAC アドレスとオプションの MAC マスクのリストです。MAC マスクはワイルドカード マスクで、ルート マップが MAC リストのエントリと一致すると論理的に MAC アドレスと AND 結合されます。MAC リストによるフィルタリングでは、パケットの MAC アドレスと MAC リスト内の MAC リストが照合されます。特定の MAC アドレスが MAC リストのどのエントリとも一致しなかった場合、実質的に拒否されたものと見なされます。

MAC リストに複数のエントリを設定し、エントリと一致した MAC アドレスを許可または拒否できます。各エントリにはシーケンス番号が関連付けられています。この番号はユーザーが構成できます。Cisco NX-OS はシーケンス番号が最も小さいエントリから順番に MAC リストを評価します。Cisco NX-OS は指定された MAC アドレスと最初に一致するエントリを処理します。一致すると、Cisco NX-OS は permit 文または deny 文を処理し、MAC リストの残りのエントリは評価しません。

ルート マップ

ルート マップは、ルートの再配布に使用できます。ルート マップ エントリは、一致基準および設定基準のリストからなります。一致基準では、着信ルートまたはパケットの一致条件を指定します。設定基準では、一致基準を満たした場合のアクションを指定します。

同じルートマップに複数のエントリを設定できます。これらのエントリには、同じルートマップ名を指定し、シーケンス番号で区別します。

一意のルート マップ名の下に 1 つまたは複数のルート マップ エントリをシーケンス番号に従って並べ、ルートマップを作成します。ルートマップエントリのパラメータは、次のとおりです。

- シーケンス番号
- アクセス権：許可または拒否
- 一致基準
- 設定変更

ルートマップではデフォルトで、最小のシーケンス番号から順にルートまたはIPパケットが処理されます。**continue** 文を使用すると、次に処理するルート マップ エントリを決定できるので、別の順序で処理するようにルート マップを設定できます。

ルートマップのシーケンスのデフォルトアクション

ルート マップ内の任意のシーケンスのデフォルト アクションは**permit.**です。許可アクションは次の状況で適用されます。

- **permit**または**deny.**を明示的に指定せずにルート マップに新しいシーケンスを設定する場合
- ルート マップで設定されたシーケンスを編集し、アクションを指定しない場合。この状況では、編集されたルートマップに元々 **deny.** が設定されていた場合でも、**permit** アクションが適用されます。たとえば、シーケンス 10 が deny で設定されていると仮定します。後ほど、**deny** を再度指定せずにシーケンス 10 を編集すると、そのシーケンスのアクションは **permit.** に設定されます。

ルートマップのシーケンスを設定または編集する場合は、常に正しいアクションを設定してください。そうしないと、デフォルトのアクションである **permit** が適用されます。

ルート マップのデフォルト シーケンス番号

シーケンス値が指定されていないルートマップのデフォルトのシーケンス番号は10です。シーケンス番号を指定せずに新しいルートマップを作成すると、新しいルートのシーケンス番号はデフォルトで 10 になります。デフォルトのシーケンス番号は、次のような状況でも適用されます。

- **シーケンス番号 10 の既存のルートマップ：シーケンス番号 10：**ルートマップがすでに存在し、シーケンス番号を指定せずに同じルートマップを再度構成した場合、変更はそのルートマップのシーケンス番号 10 に適用されます。
- **他のシーケンス番号（20、30、40 など）を持つ既存のルートマップ：**ルートマップにすでにシーケンス番号（20、30、40 など）が割り当てられており、シーケンス番号を指定せずに再度構成する場合、そのルートマップに対してシーケンス番号10の新しいエントリが作成されます。

一致基準

さまざまな基準を使用して、ルート マップでルートや IP パケットを照合できます。BGP コミュニティ リストのように、特定のルーティング プロトコルだけに適用できる基準もありますが、IP 送信元または宛先アドレスなど、その他の基準はあらゆるルートまたはIPパケットに使用できます。

ルートマップに従ってルートまたはパケットを処理する場合、Cisco NX-OS は設定されている個々の match 文とルートまたはパケットを比較します。ルートまたはパケットが設定されている基準と一致した場合、Cisco NX-OS はルート マップ内で一致するエントリに対する許可または拒否設定、および設定されている設定基準に基づいて、このルートやパケットを処理します。

一致のカテゴリおよびパラメータは、次のとおりです。

- BGP パラメータ：AS 番号、AS パス、コミュニティ属性、または拡張コミュニティ属性に基づく一致
- プレフィックス リスト：アドレスまたはアドレス範囲に基づく一致
- マルチキャスト パラメータ：ランデブー ポイント、グループ、または送信元に基づく一致
- その他のパラメータ：IP ネクストホップ アドレスまたはパケット長に基づく一致

設定変更

ルートまたはパケットがルート マップのエントリと一致したら、設定済みの 1 つ以上の set 文に基づいて、そのルートまたはパケットを変更できます。

設定変更は次のとおりです。

- BGP パラメータ：AS パス、タグ、コミュニティ、拡張コミュニティ、ダンプニング、ローカル プリファレンス、オリジン、または重み値属性の変更
- メトリック：ルート メトリックまたはルート タイプの変更
- その他のパラメータ：フォワーディング アドレスまたは IP ネクストホップ アドレスの変更

アクセス リスト

IP アクセス リストでは、次のような IP パケット フィールドとパケットを照合できます。

- 送信元または宛先 IPv4 または IPv6 アドレス
- プロトコル
- Precedence
- ToS
- ルート マップで ACL（アクセス コントロール リスト）を使用できるのは、ポリシーベース ルーティングの場合に限られます。

BGP の AS 番号

BGP ピアとの照合に使用する AS 番号のリストを設定できます。BGP ピアがリスト内の AS 番号と一致し、さらに他の BGP ピア設定と一致する場合、BGP はセッションを作成します。BGP ピアがリスト内の AS 番号と一致しない場合は、BGP はピアを無視します。AS 番号は AS 番号の範囲のリストとして設定できます。また、AS パス リストを使用して AS 番号を正規表現と比較することもできます。

BGP の AS パス リスト

AS パス リストを設定すると、着信または発信 BGP ルートのアップデートをフィルタリングできます。ルート アップデートに AS パス リストのエントリと一致する AS パス属性が含まれている

場合、ルータは設定されている許可または拒否条件に基づいてルートを処理します。ルート マップの中で AS パス リストを設定できます。

同じ AS パス リスト名を使用することによって、AS パス リストで複数の AS パス エントリを設定できます。ルータは最初に一致したエントリを処理します。

BGP のコミュニティ リスト

ルート マップのコミュニティ リストを使用すると、BGP コミュニティに基づいて BGP ルート アップデートをフィルタリングできます。コミュニティ属性はコミュニティ リストに基づいて照合できます。また、コミュニティ属性はルート マップを使用して設定できます。

コミュニティ リストには、1 つまたは複数のコミュニティ属性を指定します。同じコミュニティ リスト エントリに複数のコミュニティ属性を設定した場合、BGP ルートが一致と見なされるには、指定されたすべてのコミュニティ属性と一致しなければなりません。

同じコミュニティリスト名を使用することによって、コミュニティリストのそれぞれ個別のエントリとして、複数のコミュニティ属性を設定することもできます。この場合、ルータは最初に BGP ルートと一致したコミュニティ属性を、そのエントリの許可または拒否設定に基づいて処理します。

コミュニティ リストのコミュニティ属性は、次の形式のいずれか 1 つで設定できます。

- 名前付きコミュニティ属性（**internet**、**no-export** など）。
- *aa:nn* 形式（最初の 2 バイトは 2 バイトの自律システム番号、最後の 2 バイトはユーザが定義するネットワーク番号を表します）。
- 正規表現。

BGP の拡張コミュニティ リスト

拡張コミュニティ リストでは 4 バイトの AS 番号がサポートされています。拡張コミュニティ リストのコミュニティ属性は、次のいずれかの形式で設定できます。

- *aa4:nn* 形式（最初の 4 バイトは 4 バイトの AS 番号、最後の 2 バイトはユーザが定義するネットワーク番号を表します）。
- 正規表現。

Cisco NX-OS は汎用の特定拡張コミュニティ リストをサポートしています。このリストを使用すると、4 バイトの AS 番号に対して通常のコミュニティ リストと同様の機能を使用できます。汎用の特定拡張コミュニティ リストには次のプロパティを設定できます。

- Transitive：BGP はコミュニティ属性を自律システム間に伝達します。
- Nontransitive：BGP はコミュニティ属性を削除してからルートを他の自律システムに伝達します。

NX-OS BGP の大規模コミュニティの構成

NX-OS BGP の大規模コミュニティについて

NX-OS BGPは、標準および拡張コミュニティのみをサポートします。各標準コミュニティにはそれぞれ 4 バイトの制限があり、拡張コミュニティには 8 バイトの制限があるため、4 バイト ASN の使用はルートの分類方法に制限されます。8 バイトのうち、2 バイトはコミュニティ タイプの定義に使用されますが、残りの 6 バイトは使用可能です。大規模コミュニティは、IETF RFC（8092）によって標準化されています。これによりサイズが 12 バイトの大規模なコミュニティを定義でき、BGP ルートの分類が柔軟に行えます。

この機能は、コミュニティを使用してルートにタグを付けることにより、異なるASN内の異なるデータ センターからのルートを分類する機能を提供します。大規模コミュニティは、それぞれが12バイトの長さであるため、さまざまなASNからのルートを分類する役割を果たせます。RFC8092のサポートを追加することにより、NX-OS BGP は、標準のルート ポリシー メソッドを使用して 4 バイト ASN からルートを分類することを可能にします。また、標準の BGP コミュニティの 4 バイトの制限を取り除くことにより、ネットワークとルーティング ポリシーをより柔軟に構成できるようになります。

大規模なコミュニティ リストの構成（拡張）

大規模コミュニティ リストを展開形式で構成する手順は次のとおりです。

手順の概要

1. **configure terminal**
2. **ip large-community-list** *expanded*
3. **ip large-community-list expanded** *list-name*
4. **ip large-community-list expanded abcd** *seq*
5. **ip large-community-list expanded abcd seq 10** {*deny* | *permit*}
6. **ip large-community-list expanded abcd seq 10 permit** *XX:YY:ZZ*

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**ip large-community-list** *expanded* 例： `switch(config)# ip large-community-list expanded`	このオプションは、拡張された大規模コミュニティ リスト エントリを追加します。

	コマンドまたはアクション	目的
Step 3	**ip large-community-list expanded** *list-name* 例： `switch(config)# ip large-community-list expanded list-name`	このオプションは、拡張された大規模コミュニティリストの名前を提供します。*list-*名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。
Step 4	**ip large-community-list expanded abcd** *seq* 例： `switch(config)# ip large-community-list expanded abcd seq`	このオプションは、エントリのシーケンス番号を提供します。
Step 5	**ip large-community-list expanded abcd seq 10** {*deny* | *permit*} 例： `switch(config)# ip large-community-list expanded abcd seq 10 {deny | permit}`	最初のオプションは、拒否する大規模なコミュニティを指定します。 2 番目のオプションは、受け入れる大規模なコミュニティを指定します。
Step 6	**ip large-community-list expanded abcd seq 10 permit** *XX:YY:ZZ* 例： `switch(config)# ip large-community-list expanded abcd seq 10 permit XX:YY:ZZ`	このオプションは、XX:YY:ZZ 形式を使用する正規表現を提供します。XX の範囲は <0 ～ 4294967294> で、ASN を表す 4 オクテットのグローバル管理者フィールドです。一方、YY と ZZ は 4 オクテットのローカルデータ フィールドであり、ASN の所有者によって定義されます。 ":" は、グローバルデータ フィールドとローカルデータ フィールドの間の区切り文字です。

例

次の例は、拡大形式の大規模コミュニティ リストの作成方法を示しています：

```
switch(config)# ip large-community-list expanded abcd seq 10 permit ”^100:200:300$"
switch(config)# sh run rpm
<<SNIP>>
ip large-community-list expanded abcd seq 10 permit ”^100:200:300$"
```

大規模なコミュニティ リストの構成（標準）

大規模なコミュニティ リストを標準形式で構成する手順は次のとおりです。

手順の概要

1. **configure terminal**
2. **ip large-community-list** *standard*
3. **ip large-community-list standard** *list-name*
4. **ip large-community-list standard efgh** *seq*

5. **ip large-community-list standard efgh seq 15** {*deny* | *permit*}
6. **ip large-community-list standard efgh seq 15 deny** *XX:YY:ZZ*

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**ip large-community-list** *standard* 例： `switch(config)# ip large-community-list standard`	このオプションは、標準の大規模なコミュニティ リスト エントリを追加します。
Step 3	**ip large-community-list standard** *list-name* 例： `switch(config)# ip large-community-list standard list-name`	このオプションは、標準の大規模なコミュニティ リストの名前を提供します。*list*-名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。

Step 4	**ip large-community-list standard efgh** *seq*
	例：
	`switch(config)# ip large-community-list standard efgh`
	`seq`
		このオプションは、エントリのシーケンス番号を提供します。

Step 5	**ip large-community-list standard efgh seq 15** {*deny* | *permit*}
	例：
	`switch(config)# ip large-community-list standard efgh seq 15`
	`{deny | permit}`
		最初のオプションは、拒否する大規模なコミュニティを指定します。
		2 番目のオプションは、受け入れる大規模なコミュニティを指定します。

Step 6	**ip large-community-list standard efgh seq 15 deny** *XX:YY:ZZ*
	例：
	`switch(config)# ip large-community-list standard efgh seq 15 deny`
	`XX:YY:ZZ`
		このオプションは、XX:YY:ZZ 形式を使用する正規表現を提供します。XX の範囲は <0 ～ 4294967294> で、ASN を表す 4 オクテットのグローバル管理者フィールドです。一方、YY と ZZ は 4 オクテットのローカルデータ フィールドであり、ASN の所有者によって定義されます。
		":" は、グローバルデータ フィールドとローカルデータ フィールドの間の区切り文字です。

例

次の例は、標準形式の大規模なコミュニティ リストの作成方法を示しています：

```
switch(config-route-map)# ip large-community-list standard efgh seq 15 deny 1000300:123:456
switch(config)# sh run rpm
<<SNIP>>
ip large-community-list standard efgh seq 15 deny 1000300:123:456
```

大規模コミュニティのルート マップ マッチの構成

大規模コミュニティのルート マップ マッチを構成する手順は次のとおりです。

手順の概要

1. **configure terminal**
2. **match** *large-community*
3. **match large-community** *list-name*
4. **match large-community abcd** *exact-match*

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**match** *large-community* 例： `switch(config-route-map)# match large-community`	このオプションは、BGP 大規模コミュニティ リストとのマッチングを行います。
Step 3	**match large-community** *list-name* 例： `switch(config-route-map)# match large-community list-name`	このオプションは、コミュニティ リストの名前を提供します。*list*-名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。
Step 4	**match large-community abcd** *exact-match* 例： `switch(config-route-map)# match large-community abcde exact-match`	このオプションは、コミュニティの完全一致を実行します。

例

次の例は、拡大形式の大規模コミュニティ リストの作成方法を示しています：

```
switch(config-route-map)# sh run rpm
<<SNIP>>
route-map test permit 10
  match large-community abcd efgh
```

大規模コミュニティのルート マップ セットの構成

大規模コミュニティのルート マップ セットを構成する手順は次のとおりです。

手順の概要

1. **configure terminal**
2. **set** 大コミュニティリスト
3. **set large-community-list** *list-name*
4. **set large-community-list list-name** *delete*
5. { なし | XX:YY:ZZ [添加剤] |添加物}**set large-community**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**set** 大コミュニティリスト 例： `switch(config-route-map)# set large-community-list`	このオプションは、BGP 大規模コミュニティ属性を設定します。
Step 3	**set large-community-list** *list-name* 例： `switch(config-route-map)# set large-community-list list-name`	このオプションは、大規模コミュニティ リストの名前を設定します。*list-*名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。
Step 4	**set large-community-list list-name** *delete* 例： `switch(config-route-map)# set large-community-list list-name delete` 例：	このオプションは、マッチした大規模なコミュニティを削除します。

	コマンドまたはアクション	目的

	`switch(config-route-map)# sh run rpm`
	`route-map test permit 10`
	`set large-community-list list-name delete`	

Step 5	{ なし | XX:YY:ZZ [添加剤] |添加物}**set large-community**
	例:
	`switch(config-route-map)# set large-community {none | XX:YY:ZZ [additive] | additive}`
	`switch(config-route-map)# set large-community 1000:1235:7629 200:30048:234 additive`
	例:
	`switch(config-route-map)# sh run rpm`
	`route-map test permit 10`
	`set large-community additive`
	`switch(config-route-map)# sh run rpm`
	`route-map test permit 10`
	`set large-community 1000300:123:456`
	`switch(config-route-map)# sh run rpm`
	`route-map test permit 10`
	`set large-community none`
		このコマンドは、BGP ルート更新のための大規模コミュニティ属性を設定します
		• 「XX:YY:ZZ」オプションは、XX:YY:ZZ 形式で表した大規模コミュニティ属性を示しています。BGP ルート アップデートでは、その値のみを設定します。1 つの set コマンドで最大 32 の大規模コミュニティ属性を追加できます。
		• 「additive」オプションは、既存の大規模コミュニティ属性への追加を表しており、XX:YY:ZZ オプションとともに使用されます。この方法を使用すると、既存の大規模コミュニティ属性に XX:YY:ZZ 属性が追加されます。
		• 「なし」オプションは、大規模コミュニティ属性が設定されないことを表します。

ルートの再配布およびルート マップ

ルートマップを使用すると、ルーティング ドメイン間でのルートの再配布を制御できます。ルート マップではルートの属性を照合し、一致基準を満たすルートだけを再配布します。設定変更を使用することによって、再配布時に、ルート マップでルート属性を変更することもできます。

ルータは再配布されたルートを各ルート マップ シーケンスと照合します。ルートマップ シーケンスで match 文が複数ある場合は、ルートマップ シーケンスでルートがすべての一致基準を満たしている必要があります。ルートがルートマップ シーケンスで定義されている一致基準を満たす場合は、そのシーケンスで拒否されたセットアクションが実行されます。ルートマップ シーケンスでルートが基準と一致しなかった場合、ルータは後続のルートマップ シーケンスとルートを比較します。ルートマップに対するこのルート評価は、一致するか、ルート マップ内のすべてのシーケンスによってルートが評価されるまで継続されます。最後に、ルートがルートマップシーケンスのいずれとも一致しない場合、ルータはルートの受け入れを拒否するか（インバウンド ルートマップの場合）、ルートの転送を拒否します（アウトバウンド ルートマップの場合）。

（注） BGP をIGP に再配布するとき、iBGP も再配布されます。この動作を無効にするには、ルート マップに追加 deny 文を挿入します。

Route Policy Manager の注意事項と制約事項

Route Policy Manager 設定時の注意事項および制約事項は、次のとおりです。

- プレフィックス リスト内の名前は、大文字と小文字が区別されません。一意の名前を使用することを推奨します。大文字と小文字を変更しただけの同じ名前を使用しないでください。たとえば、CTCPrimaryNetworks と CtcPrimaryNetworks は 2 つの異なるエントリではありません。
- ルートマップが存在しない場合、すべてのルートが拒否されます。
- プレフィックス リストが存在しない場合は、すべてのルートが許可されます。
- ルート マップ エントリで 2 つの無関係なエンティティを照合する場合、ルート マップ エントリのアクセス権（許可または拒否）によって、すべてのルートまたはパケットの処理結果が決まります。また、ルート マップ エントリの設定基準も適用されます。たとえば、次のルート マップは、BGP 設定に関連付けられている場合、ospf-area とのマッチングを試みます。その結果、無関係なマッチングが許可されて、メトリックが 100 に設定されます。

```
route-map abc permit seq 10
match ospf-area 2
set metric 100
```

- ルート マップ エントリに match 文がない場合、ルート マップ エントリのアクセス権（許可または拒否）によって、すべてのルートまたはパケットの処理結果が決まります。
- ルートマップ エントリの match 文の中で参照されたポリシー（プレフィックス リストなど）から no-match または deny-match が戻った場合、 は match 文を Cisco NX-OS 失敗として、次のルート マップ エントリを処理します。
- ルート マップを変更しても、ルート マップ コンフィギュレーション サブモードを終了するまでは、Cisco NX-OS によりすべての変更が保留されます。その後、Cisco NX-OS がすべての変更をプロトコル クライアントに送信すると、変更が有効になります。
- 同じルートマップシーケンスに IPv4 と IPv6 の両方の match ステートメントを含めないことを推奨します。両方が必要な場合は、同じルートマップの異なるシーケンスで指定する必要があります。
- ルート マップは定義する前に使用できるので、設定変更を終えるときには、すべてのルートマップが存在していることを確認してください。
- 再配布およびフィルタリングを行う場合、ルート マップの使用状況を確認できます。各ルーティング プロトコルには、これらの統計情報を表示する機能があります。
- BGP をIGP に再配布するとき、iBGP も再配布されます。この動作を無効にするには、ルートマップに追加 deny 文を挿入します。
- Route Policy Manager は MAC リストをサポートしていません。

- ip access-list name コマンドの ACL 名の最大文字数は 64 です。ただし、RPM コマンドに関連付けられている ACL 名（ip prefix-list や match ip address など）は、最大 63 文字しか使用できません。
- BGP は特定の **match** コマンドのみをサポートします。詳細については、「ルート マップの設定」の **match** コマンドの表を参照してください。
- 「prefix-list」という名前の ACL を作成する場合、match ip address コマンドを使用して作成されたルート マップに関連付けることはできません。RPM コマンドの match ip address prefix-list は、前のコマンド（「prefix-list」ACL 名）をあいまいにします。
- match ip address コマンドを使用する場合、設定できる ACL は 1 つだけです。
- ポリシーが構成プロファイルを介して適用される場合、通常の CLI 構成モードを介して特定の CLI の構成解除（短い no 形式）を試行することは推奨されません。変更が必要な場合は、まずプロファイルの適用を解除してから、プロファイルを変更して再度適用します。
- すべての RPM プロファイルで、構成プロファイルを構成して適用する場合は、後で「構成プロファイル」を使用するときのために、同じプロファイルを構成および構成解除しないでください（短い no 形式を使用）。
- config-profile の複数の行で standard ip community-list と ip large-community-list を構成すると、そのシーケンスの最後に構成された行のみが保持されます。これら 2 つのコマンドを実行するには、すべてのコミュニティ値を構成し、config-profile で 1 つのコマンドとして実行する必要があります。
- Cisco NX-OS リリース 10.2(2)F 以降、BGP NLRI（インバウンドおよびアウトバウンドのルートマップ用）のタグの照合がサポートされるようになりました。ただし、これは、VXLAN の L4-7 サービス統合で L2VPN EVPN アドレス ファミリを使用することのみを目的としています。

Route Policy Manager パラメータのデフォルト設定

次の表に、Route Policy Manager のデフォルト設定を示します。

表 *27 :* デフォルトの *Route Policy Manager* パラメータ

パラメータ	デフォルト
Route Policy Manager	有効（Enabled）
アドミニストレーティブ ディスタンス	115

Route Policy Manager の設定

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

IP プレフィックス リストの設定

IP プレフィックス リストでは、プレフィックスおよびプレフィックス長のリストに対して IP パケットまたはルートを照合します。IPv4 には IP プレフィックス リスト、IPv6 には IPv6 プレフィックス リストを作成できます。

指定したプレフィックス長と完全に一致するプレフィックスリストエントリのみを対象とするよう設定できます。また、指定したプレフィックス長の範囲に該当するすべてのプレフィックスを対象とすることもできます。

Cisco NX-OS リリース 9.3(9) 以降、NDFC/config-profile/dual-stage 構成モードの prefix-list を構成するとき、シーケンス番号が追加されていることを確認してください。また、シーケンスを変更するか、新しいものを挿入するとき、シーケンス番号を割り当てる代わりに、できれば 5 または 10 刻みで、シーケンス番号にずれが生じるようにします。

次に例を示します。

```
ip prefix-list allowprefix seq 10 permit 192.0.2.0/23 eq 24
ip prefix-list allowprefix seq 20 permit 209.165.201.0/27 eq 28
```

（注） Cisco NX-OS リリース 9.3.9 以降、prefix-list に config-profile にシーケンス番号がない場合は、 本リリース以降にアップグレードする前にシーケンス番号を追加してください。

ge キーワードと **lt** キーワードを使用すると、プレフィックス長の範囲を指定できます。着信パケットまたはルートがプレフィックス リストと一致すると判定されるのは、プレフィックスが一致し、プレフィックス長が **ge** キーワードの値（設定されている場合）以上かつ **lt** キーワードの値（設定されている場合）以下の場合です。キーワード **eq** を使用する場合、設定する値はプレフィックスのマスク長より大きくする必要があります。

プレフィックス アドレスとの比較に使用できる連続または非連続ルートの範囲を定義するには、**mask** キーワードを使用します。

手順の概要

1. **configure terminal**
2. { **ip** | **ipv6** } **prefix-list** *name* **description** *string*
3. {**ip** | **ipv6**} **prefix-list** *name* [**seq** *number*] [{ **permit** | **deny** } *prefix* {[**eq** *prefix-length*] | [**ge** *prefix-length*] [**le** *prefix-length*]}] [**mask** *mask*]

4. （任意） **show** { **ip** | **ipv6** } **prefix-list** *name*
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	必須: { **ip** | **ipv6** } **prefix-list** *name* **description** *string* 例： `switch(config)# ip prefix-list AllowPrefix description allows engineering server`	プレフィックス リストについての情報ストリングを追加します。

Step 3	{**ip** | **ipv6**} **prefix-list** *name* [**seq** *number*] [{ **permit** | **deny** } *prefix* {[**eq** *prefix-length*] | [**ge** *prefix-length*] [**le** *prefix-length*]}] [**mask** *mask*]
	例：
	`switch(config)# ip prefix-list AllowPrefix seq 10 permit 192.0.2.0/23 eq 24`
	`switch(config)# ipv6 prefix-list AllowIPv6Prefix seq 10 permit 2001:0DB8:: le 32`
	`switch(config)# ip prefix-list even permit 0.0.0.0/32 mask 0.0.0.1`
	`switch(config)# ipv6 prefix-list even permit 2001:0DB8::/64 mask ffff:1::`
		IPv4 または IPv6 プレフィックス リストを作成するか、または既存のプレフィックスリストにプレフィックスを追加します。*prefix-length* は次のように照合されます。
		• **eq**：正確なプレフィックス長を照合します。この値は、マスク長より大きくする必要があります。
		• **ge**：設定されている *prefix length* 以上のプレフィックス長が対象。
		• **le**：設定されている *prefix length* 以下のプレフィックス長が対象。
		• **mask**：ルーティング プロトコルで使用されるプレフィックス アドレスのビットと比較する、プレフィックス リストのプレフィックス アドレスのビットを指定します。このオプションは、Cisco Nexus 9200、9300-EX、および9300-FXプラットフォームスイッチと9700-EXおよび9700-FXラインカードのCisco NX-OSリリース9.3（3）以降で使用できます。

Step 4	（任意） **show** { **ip** | **ipv6** } **prefix-list** *name* 例：	プレフィックスリストについての情報を表示します。

	コマンドまたはアクション	目的
	`switch(config)# show ip prefix-list AllowPrefix`	
Step 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、2 つのエントリからなる IPv4 プレフィックス リストを作成し、BGP ネイバーにプレフィックス リストを適用する例を示します。

```
switch# configure terminal
switch(config)# ip prefix-list allowprefix seq 10 permit 192.0.2.0/23 eq 24
switch(config)# ip prefix-list allowprefix seq 20 permit 209.165.201.0/27 eq 28
switch(config)# router bgp 65535
switch(config-router)# neighbor 192.0.2.1/16 remote-as 65534
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# prefix-list allowprefix in
```

次に、すべての/ 24奇数IPアドレスの一致マスクを使用してIPv4プレフィックスリストを作成する例を示します。

```
switch# configure terminal
switch(config)# ip prefix-list list1 seq 7 permit 22.1.1.0/24 mask 255.255.1.0
switch(config)# show route-map test
route-map test, permit, sequence 7
Match clauses:
ip address prefix-lists: list1
Set clauses:
extcommunity COST:igp:10:20
switch(config)# show ip prefix-list list1
ip prefix-list list1: 1 entries
seq 7 permit 22.1.1.0/24 mask 255.255.1.0
```

次に、サブネットプレフィックスが17以上の21.1.0.0/16のすべてのサブネットに一致するIPv4プレフィックスリストを作成する例を示します。maskオプションにより、3番目のオクテットの最初のビットが設定されていない（偶数）着信プレフィックスだけが照合されます。

```
switch# configure terminal
switch(config)# ip prefix-list list1 seq 10 permit 21.1.0.0/16 ge 17 mask 255.255.1.0
```

MAC リストの設定

MAC リストを設定すると、特定の範囲の MAC アドレスを許可または拒否できます。

Cisco NX-OS リリース 10.4(2)F 以降、NDFC/config-profile/dual-stage 構成モードの prefix-list を構成するとき、シーケンス番号が追加されていることを確認してください。また、シーケンスを変更

するか、新しいものを挿入するとき、シーケンス番号を割り当てる代わりに、できれば 5 または 10 刻みで、シーケンス番号にずれが生じるようにします。

次に例を示します。

```
mac-list AllowMac seq 5 permit 0022.5579.a4c1 ffff.ffff.0000
mac-list AllowMac seq 10 permit 0033.5510.a4c1 ffff.ffff.0000
```

（注） Cisco NX-OS リリース 10.4(2)F 以降、mac-list に config-profile にシーケンス番号がない場合は、本リリース以降にアップグレードする前にシーケンス番号を追加してください。

手順の概要

1. **configure terminal**
2. mac-list *name* seq *number* {permit | deny } *mac-address* [*mac-mask*]
3. （任意） **show mac-list** *name*
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例:
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	必須: mac-list *name* seq *number* {permit | deny } *mac-address* [*mac-mask*] 例: `switch(config)# mac-list AllowMac seq 5 permit 0022.5579.a4c1 ffff.ffff.0000`	MAC リストを作成するか、既存の MAC リストに MAC アドレスを追加します。seq の範囲は 1 ～ 4294967294 です。*mac-mask* は照合する MAC アドレスの部分を表します。MAC アドレス形式である必要があります。
Step 3	（任意） **show mac-list** *name* 例: `switch(config)# show mac-list name`	プレフィックスリストについての情報を表示します。
Step 4	（任意） **copy running-config startup-config** 例: `switch(config)# copy running-config startup-config`	この設定変更を保存します。

AS パス リストの設定

発信と着信の両方の BGP ルートに AS パス リスト フィルタを指定できます。各フィルタは、正規表現を使用するアクセス リストです。正規表現が ASCII ストリングとして表されたルートの AS パス属性と一致した場合は、許可または拒否条件が適用されます。

手順の概要

1. **configure terminal**
2. **ip as-path access-list** *name* {**deny** | **permit**} *expression*
3. （任意） **show** {**ip** | **ipv6**} **as-path-access-list** *name*
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**ip as-path access-list** *name* {**deny** | **permit**} *expression* 例： `switch(config)# ip as-path access-list Allow40 permit 40`	正規表現を使用して BGP AS パス リストを作成します。
Step 3	（任意） **show** {**ip** | **ipv6**} **as-path-access-list** *name* 例： `switch(config)# show ip as-path-access-list Allow40`	as-path アクセス リストの情報を表示します。
Step 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、2 つのエントリからなる AS パス リストを作成し、BGP ネイバーに AS パス リストを適用する例を示します。

```
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 65535:20
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
```

BGP AS-path 属性の置き換え

次の手順では、着信および発信ルートマップの BGP as-path 属性を変更することにより、BGP ルーティング ポリシーを操作できます。

BGP as-path 属性を置き換えるときは、次のガイドラインを考慮してください。

- この機能は、アドレスファミリ識別子 (AFI) ごとに eBGP ネイバーにのみ適用されます。iBGP ネイバーで機能を設定しようとしても、構成は無視されます。
- この機能を備えたルートマップは、BGP ネイバーのインバウンド側とアウトバウンド側の両方に適用できます。
- この機能は、AS_SET、AS_SEQUENCE、CONFED_SET、および CONFED_SEQUENCE の任意の組み合わせをサポートします。
- 2 バイト AS のみをサポートする BGP スピーカーと対話する場合、4 バイト AS 番号は予約済みの 2 バイト AS 番号 23456 に置き換えられます。
- コンフェデレーション識別子が設定されている場合は、コンフェデレーションの外部にあるピアと対話するときに、CLI でローカル ASN としてコンフェデレーション識別子を使用することを検討してください。同じコンフェデレーションに属するピアと対話する場合は、**router bgp** *asn* コマンドでプロセス ASN を使用することを検討してください。
- BGP **local-as** 機能が設定されている場合、設定された local-as は CLI でローカル ASN と見なされます。
- アウトバウンド ルートマップの場合、ローカル ASN は常に CLI からの結果の as_path に付加されます。
- **set as-path** または **set as-path replace** コマンドでは、最大 32 個の AS 番号を設定できます。
- 1 つのルートマップ シーケンスの下では、**set as-path**、**set as-path prepend**、および **set as-path replace** のオプションのうち 1 つだけを設定できます。
- **remove-private-as** が設定されている場合、アウトバウンド側で新しいルートマップ コマンドを適用する前に適用されます。
- **as-override** が設定されている場合、アウトバウンド側で新しいルートマップ コマンドを適用した後に適用されます。
- AS_PATH ループ チェックは、新しいルートマップ コマンドが着信側と発信側の両方に適用される前に、元の AS_PATH で実行されます。これらのチェックは、インバウンド側で **allow-as in** とアウトバウンド側で**disable-peer-as-check** を使用することで緩和できます。

完全な AS パスの置き換え

この手順を使用して、着信または発信 BGP アップデートの AS パスをカスタム AS パスに変更します。AS パスを完全に削除することもできます。

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**route-map** *map-name* [**permit** | **deny**] [*seq*]
	例：
	`switch(config)# route-map Testmap permit 10`
	`switch(config-route-map)#`
		ルート マップを作成するか、または既存のルート マップに対応するルート マップ設定モードを開始します。ルート マップのエントリを順序付けるには、*seq* を使用します。

Step 3	[**no**] **set as-path** { **none** | {*as-number* | **remote-as** | **local-as**}+] } 例： `switch(config-route-map)# set as-path 11 local-as remote-as 13`	AS_PATH をカスタム ASN のリストに置き換えるか、AS_PATH をクリアします。コマンド オプションは次のとおりです。 • *as-number*: 指定された AS 番号。 • **remote-as**: BGP ピアの AS 番号。 • **local-as**: ローカル AS 番号。 **none** キーワードは、AS パスを完全に削除します。

例

次の例では、これらの値が想定されています。

- 元の AS_PATH は **10 20 30 40 50 60** です。
- local-as は **100** です。
- remote-as は **200** です。

この例は、カスタム AS パスを指定する方法を示しています。このコマンドは、AS パスを **11 100 200 13 200 10.10 65535** に変更します。

```
switch# configure terminal
switch(config)# route-map Testmap permit 10
switch(config-route-map)# set as-path 11 local-as remote-as 13 remote-as 10.10 65535
```

この例は、AS パスをクリアする方法を示しています。このコマンドにより、AS パスが空になります。

```
switch# configure terminal
switch(config)# route-map Testmap permit 10
switch(config-route-map)# set as-path none
```

AS パスでの選択した AS 番号の置き換え

この手順を使用して、AS パス内の特定の AS 番号を置き換え、着信または発信 BGP 更新でそれらをカスタム AS 番号に置き換えます。**private-as** をマッチ キーワードとして指定することもできます。この場合、private-as の任意のインスタンスが一致し、置換または削除できます。

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**route-map** *map-name* [**permit** | **deny**] [*seq*]
	例：
	`switch(config)# route-map Testmap permit 10`
	`switch(config-route-map)#`
		ルート マップを作成するか、または既存のルート マップに対応するルート マップ設定モードを開始します。ルート マップのエントリを順序付けるには、*seq* を使用します。

Step 3	[**no**] **set as-path replace** {*asn_list* | **private-as**} [**with** {*as-number* | **remote-as** | **none**}] 例： `switch(config-route-map)# set as-path replace 1, 2, private-as with remote-as`	**with** キーワードが指定されていない場合は、コンマで区切られた *asn_list* で示されている ASN のインスタンスを local-as に置き換えます。**private-as** キーワードが指定されている場合は、private-as を置き換えます。 **with** キーワードが指定されている場合は、一致した ASN の **with** キーワードの後の値、または **private-as** キーワードが指定されている場合は private-as を置き換えます。 **with** キーワードに続くコマンド オプションは次のとおりです。 • *as-number* : 一致した値は、指定された AS 番号に置き換えられます。 • **remote-as**: 一致した値は、BGP ピアの AS 番号に置き換えられます。 • **none**: 一致した値は AS-path から削除されます。

例

次の例では、これらの値が想定されます。

- 元の AS_PATH は **1 5 2 10.10 65534 20** です。
- local-as は **100** です。
- remote-as は **200** です。

この例は、2 つの特定の ASN と、private-as を local-as に置き換える方法を示しています。このコマンドは、AS パスを **100 5 100 10.10 100 20** に変更します。

```
switch# configure terminal
switch(config)# route-map Testmap permit 10
switch(config-route-map)# set as-path replace 1, 2, private-as
```

この例は、2 つの特定の ASN と、private-as をネイバーの ASN (remote-as) に置き換える方法を示しています。このコマンドは、AS パスを **200 5 200 10.10 200 20** に変更します。

```
switch# configure terminal
switch(config)# route-map Testmap permit 10
switch(config-route-map)# set as-path replace 1, 2, private-as with remote-as
```

この例は、2 つの特定の ASN と private-as を削除する方法を示しています。このコマンドは、AS パスを **5 10.10 20** に変更します。

```
switch# configure terminal
switch(config)# route-map Testmap permit 10
switch(config-route-map)# set as-path replace 1, 2, private-as with none
```

コミュニティ リストの設定

コミュニティリストを使用すると、コミュニティ属性に基づいてBGPルートをフィルタリングできます。コミュニティ番号は *aa:nn* 形式の 4 バイト値です。最初の 2 バイトは自律システム番号を表し、最後の 2 バイトはユーザ定義のネットワーク番号です。

同じコミュニティ リスト文で複数の値を設定した場合、コミュニティ リスト フィルタを満足させるには、すべてのコミュニティ値が一致しなければなりません。複数の値をそれぞれ個別のコミュニティ リスト文で設定した場合は、最初に条件が一致したリストが処理されます。

コミュニティリストを match 文で使用すると、コミュニティ属性に基づいて BGP ルートをフィルタリングできます。

手順の概要

1. **configure terminal**
2. 次のいずれか 1 つを入力します。

- **ip community-list standard** *list-name* {**deny** | **permit**} [*community-list*] [**internet**] [**local-AS**] [**no-advertise**] [**no-export**] [**graceful-shutdown**] [**blackhole**]

  または

- **ip community-list expanded** *list-name* {**deny** | **permit**} *expression*

3. （任意） **show ip community list** *name*
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	次のいずれか 1 つを入力します。
	• **ip community-list standard** *list-name* {**deny** | **permit**} [*community-list*] [**internet**] [**local-AS**] [**no-advertise**] [**no-export**] [**graceful-shutdown**] [**blackhole**]
	または
	• **ip community-list expanded** *list-name* {**deny** | **permit**} *expression*
	例：
	`switch(config)# ip community-list standard BGPCommunity permit no-advertise 65535:20`
	または
	`switch(config)# ip community-list expanded BGPComplex deny 50000:[0-9][0-9]`
		最初のオプションでは、標準 BGP 拡張コミュニティ リストを作成します。*list*-名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。*community-list* には、1 つ以上のコミュニティを *aa:nn* 形式で指定できます。
		二番目のオプションでは、正規表現を使用して BGP 拡張コミュニティ リストを作成します。

Step 3	（任意） **show ip community list** *name* 例： `switch(config)# show ip community-list BGPCommunity`	コミュニティ リストの情報を表示します。
Step 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、2 つのエントリからなるコミュニティ リストの作成例を示します。

```
switch# configure terminal
switch(config)# ip community-list standard BGPCommunity permit no-advertise 65535:20
switch(config)# ip community-list standard BGPCommunity permit local-AS no-export
switch(config)# copy running-config startup-config
```

拡張コミュニティ リストの設定

拡張コミュニティリストを使用すると、コミュニティ属性に基づいてBGPルートをフィルタリングできます。コミュニティ番号は *aa4:nn* 形式の 6 バイト値です。最初の 4 バイトは自律システム番号を表し、最後の 2 バイトはユーザ定義のネットワーク番号です。

同じ拡張コミュニティ リスト文で複数の値を設定した場合、拡張コミュニティ リスト フィルタの条件を満たすには、すべての拡張コミュニティ値が一致しなければなりません。複数の値をそれぞれ個別の拡張コミュニティ リスト文で設定した場合は、最初に条件が一致したリストが処理されます。

拡張コミュニティリストを match 文で使用すると、拡張コミュニティ属性に基づいてBGPルートをフィルタリングできます。

（注） **extcommunity** を AS2:NN または AS4:NN（as-plain）形式で常に設定します。

NX-OS リリース 10.4(3)F 以降では、 **extcommunity** を AS.dot 形式で構成できます。

手順の概要

1. **configure terminal**
2. 次のいずれか 1 つを入力します。
   - **ip extcommunity-list standard** *list-name* {**deny** | **permit**} **seq** *5* **4byteas-generic** {**transitive** | **nontransitive**} *community1* [*community2...*] **rt** *2:2* **soo** *3:3*

     または

   - **ip extcommunity-list expanded** *list-name* **seq** *5* {**deny** | **permit**} *expression*
3. **ip extcommunity-list standard** *commext* **seq** *5* **permit 4byteas-generic transitive** *1:1* **rt** *2:2* **soo** *3:3*
4. （任意） **show ip community-list** *name*
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	次のいずれか 1 つを入力します。
	• **ip extcommunity-list standard** *list-name* {**deny** | **permit**} **seq** *5* **4byteas-generic** {**transitive** | **nontransitive**} *community1* [*community2...*] **rt** *2:2* **soo** *3:3*
	または
	• **ip extcommunity-list expanded** *list-name* **seq** *5* {**deny** | **permit**} *expression*
	例：
	`switch(config)# ip extcommunity-list standard BGPExtCommunity seq 5 permit 4byteas-generic transitive 65535:20 rt 2:2 soo 3:3`
	または
	`switch(config)# ip extcommunity-list expanded BGPExtComplex seq 5 deny 1.5:[0-9][0-9]`
		最初のオプションでは、標準 BGP 拡張コミュニティ リストを作成します。*community* には、1 つ以上の拡張コミュニティを *aa4:nn* 形式で指定できます。
		二番目のオプションでは、正規表現を使用して拡張 BGP 拡張コミュニティ リストを作成します。

Step 3	**ip extcommunity-list standard** *commext* **seq** *5* **permit 4byteas-generic transitive** *1:1* **rt** *2:2* **soo** *3:3* 例： `switch(config)# ip extcommunity-list standard commext seq 5 permit 4byteas-generic transitive 1:1 rt 2:2 soo 3:3`	シーケンス番号が CLI の入力パラメータとして追加されます。 以降、extcommunity リストの設定時に入力シーケンス番号を入力する必要があります。 （注） config replace の場合、ユーザ設定ファイルには、デバイスから収集された有効な実行コンフィギュレーションが含まれている必要があります。任意の NX-OS イメージラベルを実行しているデバイスから収集できます。手動で改ざんされていない有効なファイルである必要があります。
Step 4	（任意） **show ip community-list** *name* 例： `switch(config)# show ip community-list BGPCommunity`	拡張コミュニティ リストの情報を表示します。

	コマンドまたはアクション	目的
Step 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、汎用の特定拡張コミュニティ リストを作成する例を示します。

```
switch# configure terminal
switch(config)# ip extcommunity-list standard test1 seq 5 permit 4byteas-generic transitive
65535:40 65535:60
switch(config)# copy running-config startup-config
```

ルート マップの設定

ルートマップを使用して、ルートの再配布やルートフィルタリングを行うことができます。ルート マップには、複数の一致基準と複数の設定基準を含めることができます。

BGP にルートマップを設定すると、BGP ネイバーセッションの自動ソフトクリアまたはリフレッシュのトリガーになります。

手順の概要

1. **configure terminal**
2. **route-map** *map-name* [**permit** | **deny**] [*seq*]
3. （任意） **continue** *seq*
4. （任意） **exit**
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**route-map** *map-name* [**permit**	**deny**] [*seq*]
		例：
		`switch(config)# route-map Testmap permit 10`
		`switch(config-route-map)#`
			ルート マップを作成するか、または既存のルート マップに対応するルート マップ設定モードを開始します。*seq* を使用して、ルート マップ エントリを順序付けます。

	コマンドまたはアクション	目的
Step 3	（任意） **continue** *seq* 例： `switch(config-route-map)# continue 10`	ルート マップで次を処理するシーケンス文を決定します。使用するのは、フィルタリングおよび再配布の場合だけです。
Step 4	（任意） **exit** 例： `switch(config-route-map)# exit`	ルートマップ コンフィギュレーション モードを終了します。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config-route-map)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

ルート マップ コンフィギュレーション モードで、ルート マップに対して次のオプションの match パラメータを設定できます。

（注） **default-information originate** コマンドでは、オプションのルート マップの **match** 文は無視されます。

コマンド	目的
match as-path *name* [*name...*] 例： `switch(config-route-map)# match as-path Allow40`	1 つまたは複数の AS パス リストと照合。AS パス リストは、**ip as-path access-list** コマンドで作成します。
match as-number { *number* [,*number...*] | **as-path-list** *name* [*name...*]} 例： `switch(config-route-map)# match as-number 33,50-60`	1 つまたは複数の AS 番号または AS パス リストと照合。AS パス リストは、**ip as-path access-list** コマンドで作成します。指定できる範囲は 1 ～ 65535 です。AS パス リスト名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。
match community *name* [*name...*][exact-match] 例： `switch(config-route-map)# match community BGPCommunity`	1 つまたは複数のコミュニティ リストと照合。コミュニティ リストは、**ip community-list** コマンドで作成します。

コマンド	目的
match extcommunity *name* [*name...*][**exact-match**] 例： `switch(config-route-map)# match extcommunity BGPextCommunity`	1つまたは複数の拡張コミュニティ リストと照合。コミュニティ リストは、**ip extcommunity-list** コマンドで作成します。
match interface *interface-type number* [*interface-type number...*] 例： `switch(config-route-map)# match interface e 1/2`	設定済みのインターフェイスのいずれかからのネクストホップと照合。**?** を使用すると、サポートされているインターフェイス タイプのリストを検索できます。 （注） BGP はこのコマンドをサポートしていません。
match ip address prefix-list *name* [*name...*] 例： `switch(config-route-map)# match ip address prefix-list AllowPrefix`	1 つまたは複数の IPv4 プレフィックス リストと照合。プレフィックス リストは **ip prefix-list** コマンドを使用して作成します。
match ipv6 address prefix-list *name* [*name...*] 例： `switch(config-route-map)# match ip address prefix-list AllowIPv6Prefix`	1 つまたは複数の IPv6 プレフィックス リストと照合。プレフィックス リストは **ipv6 prefix-list** コマンドを使用して作成します。
match ip multicast [**source** *ipsource*] [[**group** *ipgroup*] [*rp iprp*]] 例： `switch(config-route-map)# match ip multicast rp 192.0.2.1`	マルチキャスト送信元、グループ、またはランデブー ポイントに基づいてIPv4マルチキャスト パケットを照合。 （注） BGP はこのコマンドをサポートしていません。
match ipv6 multicast [**source** *ipsource*][[**group** *ipgroup*] [*rp iprp*]] 例： `switch(config-route-map)# match ip multicast source 2001:0DB8::1`	マルチキャスト送信元、グループ、またはランデブー ポイントに基づいてIPv6マルチキャスト パケットを照合。 （注） BGP はこのコマンドをサポートしていません。
match ip next-hop prefix-list *name* [*name ...*] 例： `switch(config-route-map)# match ip next-hop prefix-list AllowPrefix`	1 つまたは複数の IP プレフィックス リストに対して、ルートの IPv4 ネクストホップ アドレスを照合。プレフィックス リストは **ip prefix-list** コマンドを使用して作成します。

コマンド	目的
match ipv6 next-hop prefix-list *name* [*name* ...] 例： `switch(config-route-map)# match ipv6 next-hop prefix-list AllowIPv6Prefix`	1 つまたは複数の IP プレフィックス リストに対して、ルートの IPv6 ネクストホップ アドレスを照合。プレフィックス リストは **ipv6 prefix-list** コマンドを使用して作成します。
match ip route-source prefix-list *name* [*name* ...] 例： `switch(config-route-map)# match ip route-source prefix-list AllowPrefix`	1 つまたは複数の IP プレフィックス リストに対して、ルートの IPv4 ルート送信元アドレスを照合。プレフィックス リストは **ip prefix-list** コマンドを使用して作成します。
match ipv6 route-source prefix-list *name* [*name* ...] 例： `switch(config-route-map)# match ipv6 route-source prefix-list AllowIPv6Prefix`	1 つまたは複数の IP プレフィックス リストに対して、ルートの IPv6 ルート送信元アドレスを照合。プレフィックス リストは **ipv6 prefix-list** コマンドを使用して作成します。
match metric *value* [+- *deviation.*] [*value..*] 例： `switch(config-route-map)# match metric 50 + 10`	ルート メトリック値を 1 つまたは複数のメトリック値または値の範囲と照合。メトリック範囲は +- *deviation* 引数を使用して設定します。ルート マップは次の範囲に該当するすべてのルート メトリックと照合されます。 *value - deviation* ～ *value + deviation*.
match ospf-area *area-id* 例： `switch(config-route-map)# match ospf-area 1`	OSPFv2またはOSPFv3エリアIDと一致します。 エリア ID の範囲は 0 ～ 4294967295 です。 （注） BGP はこのコマンドをサポートしていません。

コマンド	目的
match route-type *route-type* 例： `switch(config-route-map)# match route-type level 1 level 2`	ルートタイプと照合。*route-type* は、次のうちの 1 つまたは複数にできます。 • external：外部ルート（BGP、EIGRP、OSPF タイプ 1 または 2） • エリア間：OSPF エリア間ルート • internal：内部ルート（OSPF エリア内またはエリア間ルートを含む） • エリア内：OSPF のエリア内ルート • レベル 1：IS-IS レベル 1 ルート • レベル 2：IS-IS レベル 2 ルート • ローカル：ローカルで生成されたルート • nssa-external：NSSA 外部ルート（OSPF タイプ 1 または 2） • type-1：OSPF 外部タイプ 1 ルート • type-2：OSPF 外部タイプ 2 ルート （注） BGP はこのコマンドをサポートしていません。
match vlan *vlan-id* [*vlan-range*] 例： `switch(config-route-map)# match vlan 3, 5-10`	VLAN と照合。 （注） BGP はこのコマンドをサポートしていません。
match rpki { invalid | not-found | valid } 例: `switch(config-route-map)# match rpki invalid`	iBGP 学習パスの場合、着信 RPKI EXTCOMM 更新と照合します。 eBGP 学習パスの場合、 ROA データベース ルックアップから取得した検証状態と照合します。 match rpki コマンドのパラメータは次のとおりです。 • **invalid**：RPKI データベース内の無効な発信元 AS です。 • **not-found**：この origin-AS は RPKI データベースでは不明です。 • **valid**：RPKI データベース内の有効な発信元 AS です。

ルート マップ設定モードで、オプションとして、ルート マップに次の set パラメータを設定できます。

コマンド	目的
set as-path { **tag** | **prepend** { **last-as** *number* | *as-1* [*as-2*. ..]}} 例： `switch(config-route-map)# set as-path prepend 10 100 110`	BGP ルートの AS パス属性を変更します。最後の AS 番号として設定された *number* または特定の AS パス値としてのストリング（*as-1 as-2...as-n*）をプリペンドできます。
set comm-list *name* **delete** 例： `switch(config-route-map)# set comm-list BGPCommunity delete`	着信または発信 BGP ルート アップデートのコミュニティ属性から、コミュニティを削除します。コミュニティ リストは **ip community-list** コマンドを使用して作成します。
set community { **none** | **additive** | **local-AS** | **no-advertise** | **no-export** | **graceful-shutdown** | **blackhole** | *community-1* [*community-2*...]} 例： `switch(config-route-map)# set community local-AS`	BGP ルート アップデートのコミュニティ属性を設定します。 （注） ルート マップ属性の同じシーケンスで、**set community** コマンドと **set comm-list delete** コマンドを両方使用すると、設定処理より先に削除処理が実行されます。 （注） **send-community** コマンドを BGP ネイバー アドレス ファミリ コンフィギュレーション モードで使用して、BGP コミュニティ属性を BGP ピアにプロパゲートします。
set dampening *half life reuse suppress duration* 例： `switch(config-route-map)# set dampening 30 1500 10000 120`	BGP ルート ダンプニング パラメータを設定します。 • *halflife*：指定できる範囲は 1 ～ 45 分です。デフォルトは 15 です。 • *reuse*：指定できる範囲は 1 ～ 20000 秒です。デフォルトは 750 です。 • *suppress*：指定できる範囲は 1 ～ 20000 です。デフォルトは 2000 です。 • *duration*：指定できる範囲は 1 ～ 255 分です。デフォルトは 60 です。
set distance *value* 例： `switch(config-route-map)# set distance 150`	OSPFv2 または OSPFv3 のルートのアドミニストレーティブ ディスタンスを設定します。範囲は 1 ～ 255 です。

コマンド	目的
set extcomm-list *name* **delete** 例： `switch(config-route-map)# set extcomm-list BGPextCommunity delete`	着信または発信 BGP ルート アップデートの拡張コミュニティ属性から、コミュニティを削除します。拡張コミュニティ リストは **ip extcommunity-list** コマンドを使用して作成します。
set extcommunity 4byteas-generic { transitive | nontransitive }{ none | additive] *community-1* [*community-2...*]} 例： `switch(config-route-map)# set extcommunity generic transitive 1.0:30`	BGP ルート アップデートの拡張コミュニティ属性を設定します。 （注） ルート マップ属性の同じシーケンスで、**set extcommunity** コマンドと **set extcomm-list delete** コマンドを両方使用すると、設定処理より先に削除処理が実行されます。 **send-community** コマンドを BGP ネイバー アドレス ファミリ コンフィギュレーション モードで使用して、BGP コミュニティ属性を BGP ピアにプロパゲートします。
set extcommunity cost *community-id1 cost* [**igp | pre-bestpath**] [*community-id2...*]} 例： `switch(config-route-map)# set extcommunity cost 33 1.0:30`	BGP ルート アップデートのコスト コミュニティ属性を設定します。この属性は、ローカルの自律システムまたは自律連合の BGP 最良パス選択プロセスをカスタマイズすることができます。community-id の範囲は 0 ～ 255 です。*cost* の範囲は 0 ～ 4294967295 です。最も低いコストを持つパスが優先されます。コストが同じ場合は、最も低いコスト コミュニティ番号を持つパスが優先されます。 **igp** キーワードは IGP コスト比較の後にコストを比較します。**pre-bestpath** キーワードは、ベストパス アルゴリズムの他のすべてのステップの前に比較します。
set extcommunity rt *community-1* [**additive**] [*community-2.. .*]} 例： `switch(config-route-map)# set extcommunity rt 1.0:30`	BGP ルート更新の拡張コミュニティ ルート ターゲット属性を設定します。community の値は、2 バイトの AS 番号: 4 バイトのネットワーク番号、4 バイトの AS 番号: 2 バイトのネットワーク番号、または IP アドレス: 2 バイトのネットワーク番号で指定します。 **additive** キーワードは、ルート ターゲットを既存の拡張コミュニティ ルート ターゲット属性に追加するために使用します。
set forwarding-address 例： `switch(config-route-map)# set forwarding-address`	OSPF のフォワーディング アドレスを設定します。

コマンド	目的
set ip next-hop unchanged 例： `switch(config-route-map)# set ip next-hop unchanged`	不変のネクスト ホップ IP アドレスを指定します。このコマンドは、BGP IPv6-over-IPv4 ピアリングに必要です。 （注） IPv4 ネクスト ホップを使用した BGP IPv6 ユニキャスト ルートの場合、NX-OS は、BGP ネイバーに向けて構成されたアウトバウンドルートマップ内で構成された **set IPv6 next-hop unchanged** コマンドをサポートしません。
set level { backbone | level-1 | level-1-2 | level-2 } 例： `switch(config-route-map)# set level backbone`	IS-IS 用にルートをインポートするエリアを設定します。IS-IS のオプションは level-1、level-1-2、または level-2 です。デフォルトは level-1 です。
set local-preference *value* 例： `switch(config-route-map)# set local-preference 4000`	BGP ローカルプリファレンス値を設定します。範囲は 0 ～ 4294967295 です。
set metric [**+** | **-**] *bandwidth-metric* 例： `switch(config-route-map)# set metric +100`	既存のメトリック値を増減します。メトリックは Kb/s 単位です。範囲は 0 ～ 4294967295 です。
set metric *bandwidth [delay reliability load mtu]* 例 ： `switch(config-route-map)# set metric 33 44 100 200 1500`	ルート メトリック値を設定します。 メトリックは次のとおりです。 • *metric0*：帯域幅（Kb/s）。範囲は 0 ～ 4294967295 です。 • *metric1*：遅延（10 マイクロ秒単位）。 • *metric2*：信頼性。指定できる範囲は 0 ～ 255（100% の信頼性）です。 • *metric3*：ロード中。指定できる範囲は 1 ～ 255（100% のロード）です。 • *metric4*：パスの MTU。有効な範囲は 1 ～ 16777215 です。

コマンド	目的
set metric-type { external | internal | type-1 | type-2 } 例： `switch(config-route-map)# set metric-type internal`	宛先ルーティングプロトコルのメトリック タイプを設定します。オプションは次のとおりです。 external：IS-IS 外部メトリック internal：BGP の MED として IGP メトリックを使用 type-1：OSPF 外部タイプ 1 メトリック type-2：OSPF 外部タイプ 2 メトリック
set nssa-only 例： `switch(config-route-map)# set nssa-only`	P ビット セットを持たない ASBR で生成されたタイプ 7 LSA を設定します。これにより、OSPF で、タイプ 7 からタイプ 5 への LSA 変換が行われなくなります。
set origin { **egp** *as-number* | **igp** | **incomplete** } 例： `switch(config-route-map)# set origin incomplete`	BGP オリジン属性を設定します。EGP *as-number* の範囲は 0 ～ 65535 です。
set weight count 例： `switch(config-route-map)# set weight 33`	BGP ルートの重み値を設定します。範囲は 0 ～ 65535 です。
set as-path-length difference <*value*> 例: `switch(config-route-map)# set as-path-length difference 5`	不等コスト ロード バランスの最適パスと比較したパスの as-path-length の差を構成します。範囲は 1 ～ 255 です。
set metric difference <*value*> 例: `switch(config-route-map)# set metric difference 100`	不等コストロードバランスの最適パスと比較したパスのメトリック値の差を構成します。範囲は 1–65535 です。
set maximum-paths <*value*> 例: `switch(config-route-map)# set maximum-paths 5`	出力ロードバランシングのために計算およびインストールされるマルチパスの最大数を設定します。範囲は、1 ～ 64 です。

set metric-type internal コマンドは、発信ポリシーと eBGP ネイバーにのみ作用します。同じ BGP ピア発信ポリシーに **metric** コマンドと **metric-type internal** コマンドを両方設定した場合、Cisco NX-OS は **metric-type internal** コマンドを無視します。

ルートマップの削除をブロックするグローバルコマンド

このセクションでは、ルートマップの削除をブロックするグローバルコマンドの詳細について説明します。グローバル コマンドは次のとおりです。

- **system default route-map validate-applied** コマンドを使用して、ルート マップの削除のブロックを有効にします。
- **no system default route-map validate-applied** コマンドを使用して、ルート マップの削除のブロックを無効にします。
- **show running-config rpm** コマンドを使用して、デフォルト以外の構成を表示します。

（注） デフォルトでは、このコマンドはデフォルト状態です。

- **show running-config rpm all** コマンドを使用して、デフォルトの構成を表示します。

（注） デフォルトでは、このコマンドはデフォルト状態です。

（注） グローバル コマンドは、デフォルトではジェネリックです。Cisco NX-OS リリース 10.2(2)F 以降、ルートマップの削除をブロックする機能は、クライアントによって使用される場合に BGP にのみ適用されます。

Route Policy Manager の設定の確認

ポリシー マネージャ設定情報を表示するには、次の作業のいずれかを行います。

コマンド	目的
show ip community-list [*name*]	コミュニティ リストの情報を表示します。
show ip ext community-list [*name*]	拡張コミュニティ リストの情報を表示します。
show [**ip** | **ipv6**] **prefix-list** [*name*]	IPv4 または IPv6 プレフィックス リストの情報を表示します。
show route-map [*name*]	ルート マップの情報を表示します。

コマンド	目的
show route-map [*name*] **brief**	ルートマップの削除機能のブロックに関する情報と、ルートマップに関連付けられているクライアントのリストを提供します。

Route Policy Manager の設定例

次の例では、アドレス ファミリを使用して Route Policy Manager を設定し、ネイバー 209.0.2.1 からのユニキャスト ルートやマルチキャスト ルートが AllowPrefix プレフィックス リストと一致した場合に、それらのルートが承認されるようにします。

```
router bgp 64496

neighbor 172.16.0.1 remote-as 64497
  address-family ipv4 unicast
    route-map filterBGP in

route-map filterBGP
 match ip address prefix-list AllowPrefix

ip prefix-list AllowPrefix 10 permit 192.0.2.0/24
ip prefix-list AllowPrefix 20 permit 172.16.201.0/27
```

関連項目

Route Policy Manager の詳細については、次の項目を参照してください。

- 基本的 BGP の設定

第 18 章

ポリシーベース ルーティングの設定

この章は、次の項で構成されています。

- ポリシーベース ルーティングについて （623 ページ）
- ポリシーベース ルーティングの前提条件 （627 ページ）
- ポリシーベース ルーティングの注意事項と制約事項 （627 ページ）
- ポリシーベース ルーティングのデフォルト設定 （630 ページ）
- ポリシーベース ルーティングの設定 （631 ページ）
- ポリシーベース ルーティングの設定の確認 （645 ページ）
- ポリシーベース ルーティングの設定例 （646 ページ）
- ポリシーベースルーティングの関連資料 （649 ページ）

ポリシーベース ルーティングについて

ポリシーベース ルーティングを使用すると、IPv4 および IPv6 トラフィック フローに定義済みのポリシーを設定し、ルーティング プロトコルから派生したルートへの依存を弱めることができます。ポリシーベース ルーティングがイネーブルのインターフェイスで受信するすべてのパケットは、拡張パケット フィルタまたはルート マップを経由して渡されます。ルート マップでは、パケットの転送先を決定するポリシーを記述します。

ポリシーベース ルーティングには、次の機能が含まれます。

- 送信元ベースルーティング：異なるユーザセットを起点とするトラフィックをポリシールータ上のそれぞれ異なる接続を使用してルーティングします。
- QoS（Quality of Service）：ネットワークの周辺で IP パケット ヘッダーに優先または ToS（タイプ オブ サービス）値を設定することによって、またはキューイング メカニズムを利用して、ネットワークのコアまたはバックボーンでトラフィックにプライオリティを設定することによって、トラフィックを差別化します（『Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide』を参照）。
- ロード シェアリング：トラフィックの特性に基づいて、複数のパスにトラフィックを分散します。

ポリシー ルート マップ

ルートマップのエントリごとに、match 文と set 文の組み合わせが 1 つずつ含まれています。match 文では、該当するパケットが特定のポリシーを満たす基準（つまり、満たすべき条件）を定義します。set 文節で、match 基準を満たしたパケットをどのようにルーティングするかを説明します。

ルート マップ文を許可または拒否として指定できます。文の解釈は次のとおりです。

- 文に許可が指定されていて、なおかつパケットが一致基準を満たしている場合は、の set 文節が適用されます。そのアクションの 1 つに、ネクスト ホップの選択が含まれます。
- 文に拒否が指定されている場合、一致基準を満たすパケットは標準のフォワーディング チャネルを通じて送り返され、宛先ベース ルーティングが実行されます。
- 文が permit とマークされ、パケットがいずれのルート -マップ文にも一致しない場合、そのパケットは通常の転送チャネルを介して返送され、宛先ベースのルーティングが実行されます。

（注） ポリシー ルーティングは、パケットの送信元となるインターフェイスではなく、パケットを受信するインターフェイス上で指定します。

ポリシーベース ルーティングの set 基準

Cisco Nexus 9000 シリーズ スイッチは、ポリシーベース ルーティングで使用されるルートマップに対して次の **set** コマンドをサポートしています。

- **set** {**ip** | **ipv6**} **next-hop**
- **set** {**ip** | **ipv6**} **default next-hop**
- **set** {**ip** | **ipv6**} **vrf** *vrf-name* **next-hop**
- **set** {**ip** | **ipv6**} **default vrf** *vrf-name* **next-hop**
- **set interface null0**
- **set vrf** *vrf-name*

これらの **set** コマンドは、ルートマップ シーケンス内では相互に排他的です。

最初のコマンドで、IP アドレスでは、パケットの転送先である宛先へのパス上の隣接ネクスト ホップ ルータを指定します。その時点でアップの接続インターフェイスに関連付けられた最初の IP アドレスがパケットのルーティングに使用されます。

（注） 任意に、最大 32 の IP アドレスにバランシング トラフィックをロードするように、ネクスト ホップ アドレスのこのコマンドを設定できます。この場合、Cisco NX-OS は各 IP フローのすべてのトラフィックを特定の IP ネクスト ホップ アドレスに送信します。

パケットが定義された一致基準のいずれにも一致しない場合、そのパケットは標準の宛先ベースルーティング プロセスを使用してルーティングされます。

set コマンドの構成の詳細については、「ルート ポリシーの設定（635 ページ）」セクションを参照してください。

ポリシーベースルーティングのルートマップサポートマトリックス

次の表に、最新の出荷リリースを実行しているCisco Nexus 9000シリーズスイッチでのポリシーベースルーティングの設定可能なmatchおよびsetステートメントを示します。

次の凡例がテーブルに適用されます。

- [はい（Yes）]：ステートメントはポリシーベースルーティングでサポートされます。
- No：ステートメントはポリシーベースルーティングではサポートされません。
- ステートメントがポリシーベースルーティングに適用されない場合は、ステートメントの横の列にダッシュ（—）が表示されます。
- 説明が必要な場合は、適切な行/列に情報が追加されます。

表 *28*：ポリシーベース ルーティングの *SET* ルート マップ ステートメント

SET ルート マップ ステートメント	ポリシーベースルーティング（PBR）
IPv4 ネクストホップ	はい
IPv6 ネクストホップ	はい
IPv4 VRF ネクスト ホップ	はい
IPv6 VRF ネクスト ホップ	はい
デフォルト IPv4 ネクスト ホップ	はい
デフォルト IPv6 ネクスト ホップ	はい
デフォルト IPv4 vrf ネクスト ホップ	対応
デフォルト IPv6 vrf ネクスト ホップ	はい
IPv4 ネクスト ホップの可用性の確認	はい
IPv6 ネクスト ホップの可用性の確認	はい
IPv4 vrf ネクスト ホップの可用性の確認	対応
IPv6 vrf ネクスト ホップの可用性の確認	はい
デフォルトの IPv4 ネクスト ホップの可用性の確認	はい

SET ルート マップ ステートメント	ポリシーベースルーティング（PBR）
デフォルトの IPv6 ネクスト ホップの可用性の確認	はい
デフォルトの IPv4 vrf ネクスト ホップの可用性の確認	対応
デフォルトの IPv6 vrf ネクスト ホップの可用性の確認	はい
インターフェイス null0	はい
VRF	はい

ルート マップ処理ロジック

ルート マップを持つインターフェイスがパケットを受信すると、転送ロジックはシーケンス番号に従い各ルートマップ ステートメントを処理します。

ルート マップ文が route-map...permit 文の場合、パケットは **match** コマンドの基準と照合されます。このコマンドは、1つ以上のアクセスコントロールエントリ（ACE）を持つACLを参照する場合があります。パケットが ACL の許可 ACE に一致すると、ポリシーベース ルーティング ロジックは **set** コマンドがパケットで指定しているアクションを実行します。

ルート マップ文に route-map... 拒否文がある場合、パケットは一致コマンドの基準と照合されます。このコマンドは、1つ以上のACEを持つACLを参照する場合があります。パケットが ACL の許可 ACE に一致すると、ポリシーベース ルーティング プロセスが停止し、パケットはデフォルト IP ルーティング テーブルを使用してルーティングされます。

（注） **set** コマンドは、**route-map... deny** 文内部に影響しません。

- ルート マップ設定に match 文が含まれていない場合、ポリシーベース ルーティング ロジックは **set** コマンドで指定されているアクションをパケットに対して実行します。すべてのパケットは、ポリシーベースルーティングを使用してルーティングされます。
- ルートマップコンフィギュレーションがmatchステートメントを参照し、matchステートメントがアクセスコントロールエントリ（ACE）のない既存のACLまたは既存のACLを参照する場合、パケットはデフォルトルーティングテーブルを使用してルーティングされます。
- **set** { **ip** | **ipv6**} **next-hop** コマンドで指定されているネクスト ホップがダウンしているか、アクセス不能であるか、削除されている場合、パケットはデフォルトルーティングテーブルを使用してルーティングされます。

Cisco NX-OS リリース 9.2(3)以降では、**next-hop ip-address load-share** コマンドを使用して、ECMP パス上でネクスト ホップが再帰的である場合、ポリシーベース ルーティング トラフィックのバ

ランスをとることができます。この状況は、次のスイッチ、ライン カード、およびモジュールでサポートされます。

- N9K-C9372TX
- N9K-X9564TX
- N9K-X9732C-EX

すべてのネクスト ホップ ルーティング要求について、ルーティング プロファイル マネージャ（RPM）はユニキャスト ルーティング情報ベース（uRIB）を使用してそれらを解決します。また、RPM はすべての ECMP パスをプログラムするため、すべての ECMP パスを均等にロード バランシングできます。PMP over ECMP は IPv4 でのみサポートされます。

ポリシーベース ルーティングの前提条件

ポリシーベース ルーティングの前提条件は、次のとおりです。

- 有効なライセンスをインストールします。
- ポリシー ベース ルーティングを有効にする必要があります。
- インターフェイスに IP アドレスを割り当て、インターフェイスをアップにしてから、ポリシーベース ルーティング用のルート マップをインターフェイス上で適用します。

ポリシーベース ルーティングの注意事項と制約事項

ポリシーベース ルーティングに関する注意事項および制約事項は、次のとおりです。

- 9700-EX/FX ライン カードを搭載した Cisco Nexus 9500 プラットフォーム スイッチは、FIB Miss トラフィックの PBR IPv6 デフォルト ネクスト ホップをサポートしません。
- 次のスイッチは、IPv4 および IPv6 のポリシーベース ルーティングをサポートします。
  - Cisco Nexus 9200 プラットフォーム スイッチ
  - Cisco Nexus 9300-EX/FX/FX2/FX3/GX/H1/H2R プラットフォーム スイッチ
  - 9636C-R、9636C-RX、および 9636Q-R ライン カードを搭載した Cisco Nexus 9508 スイッチプロトコル ネイバーが直接接続されている場合は、明示的なホワイトリストが必要になることがあります）。
- ポリシーベース ルーティングのルート マップでは、1 つのルート マップ文に match 文を 1 つだけ指定できます。
- ポリシーベース ルーティングのルート マップでは、1 つのルート マップ文に match 文を 1 つだけ指定できます。IP SLA ポリシーベース ルーティングの詳細については、「*Cisco Nexus 9000* シリーズ *NX-OS IP SLA* 設定ガイド」を参照してください。

（注） 9636C-R、9636C-RX、および 9636Q-R ライン カードを搭載した Cisco Nexus 9508 スイッチは、IP SLA をサポートしていません。

- match コマンドで、ポリシーベース ルーティング用ルート マップの複数の ACL を参照できません。
- インターフェイスが同じ仮想ルーティング/転送（VRF）インスタンスに所属している場合は、ポリシーベース ルーティング対応のさまざまなインターフェイス間で、同じルートマップを共有できます。
- 一致基準としてプレフィックスリストを使用することはサポートされていません。ポリシーベースルーティングルートマップではプレフィックスリストを使用しないでください。
- ポリシーベースルーティングは、ユニキャストトラフィックのみをサポートします。マルチキャスト トラフィックはサポートされていません。
- ポリシーベース ルーティングは、FEX ポートの着信トラフィックでサポートされていません。
- ポリシーベース ルーティングは、Cisco Nexus 9300-EX プラットフォーム スイッチの FEX ポートではサポートされません。
- 9636C-R、9636C-RX、および 9636Q-R ライン カードを搭載した Cisco Nexus 9508 スイッチのみが、レイヤ 3 ポートチャネル サブインターフェイスを使用したポリシーベース ルーティングをサポートします。
- Cisco NX-OS リリース 10.1（2）以降、レイヤ 3 ポートチャネルサブインターフェイスを使用したポリシーベースルーティングは、Cisco Nexus 9300-X クラウドスケールスイッチでサポートされます。
- ポリシーベース ルーティングのルート マップで使用する ACL には拒否アクセス コントロール エントリ（ACE）含めることができません。
- ポリシーベースルーティングは、デフォルトのシステムルーティングモードでのみサポートされます。
- インターフェイス上に複数の機能（PBR や入力 ACL など）を設定すると、それらの機能の ACL は TCAM 最適化のためにマージされます。その結果、統計情報はサポートされません。
- VXLAN を使用する PBR の場合、load-share キーワードは必要ありません。

（注） 9700-EX/FX ライン カードを搭載した Cisco Nexus 9500 プラットフォーム スイッチは、VXLAN 経由の IPv4/IPv6 ポリシーベースルーティングをサポートします。9636C-R、9636C-RX、および 9636Q-R ライン カードを搭載した Cisco Nexus 9508 スイッチは、VXLAN を介したポリシーベース ルーティングをサポートしません。

- Cisco Nexus 9000 シリーズ スイッチはポリシーベース ACL（PBACL）をサポートしています（オブジェクト グループ ACL とも呼びます）。詳細については、『*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*』を参照してください。

（注） 9636C-R、9636C-RX、および 9636Q-R ライン カードを搭載した Cisco Nexus 9508 スイッチは、PBACL をサポートしません。

- PBR over VXLAN EVPN には、次の注意事項と制限事項が適用されます。
  - PBR over VXLAN EVPN は、Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2 プラットフォーム スイッチでのみサポートされます。
  - PBR over VXLAN は、VTEP ECMP、および **set {ip | ipv6} next-hop** *ip-address* コマンドの load-share キーワードをサポートしていません。
  - PBR over VXLAN EVPN は、 **set {ip | ipv6} vrf** *vrf-name* **next-hop** *ip-address* コマンドをサポートします。**set {ip | ipv6} vrf** *vrf-name* **next-hop** *ip-address* コマンドの複数の行を使用することにより、PBR over VXLAN EVPN は、複数のネクストホップごとに異なる VRF をサポートします。
- トンネル インターフェイス経由の PBR には、次の注意事項と制限事項が適用されます。
  - Cisco NX-OS リリース 10.3(3)F 以降、トンネル インターフェイスへの PBR ネクストホップ リダイレクトは、次の制限付きで Cisco Nexus 9000 シリーズ プラットフォーム スイッチでサポートされます。
    - **gre ip** および **ipip ip** モードのみがサポートされています。
    - ルートマップの **load-share** キーワードは、複数の構成済みネクストホップがトンネルインターフェイスと非トンネルインターフェイスの組み合わせに解決される場合はサポートされません。
    - オーバーレイ ECMP（等コストパスを持つ複数のトンネルに解決する同じネクストホップ）はサポートされていません。
- PBR 高速コンバージェンスには、次の注意事項と制限事項が適用されます。
  - PBR 高速コンバージェンスは、複数の代替ネクスト ホップで定義されたルートマップ シーケンスを持ち、ロードシェアオプションなしでネクスト ホップアベイラビリティを追跡するための SLA プローブを使用して定義されたポリシーでのみサポートされます。
  - プライマリ ホップとバックアップ ネクスト ホップの同時障害は、高速パスでは処理されません。このようなイベントでは、システムはコントロールプレーンの更新にフォールバックします。
  - PBR高速コンバージェンスは、隣接関係の損失が検出されたイベントで主にサポートされます。

- PBR高速コンバージェンスは、VXLAN経由で到達可能なネクストホップではサポートされません。
- PBR高速コンバージェンスは、可用性を追跡するためにミリ秒のSLA/トラックでネクスト ホップが指定されている場合は使用しないでください。

  SLAの設定の詳細については、『*Cisco Nexus 9000* シリーズ *NX-OS IP SLA* 設定ガイド』を参照してください。
- PBR高速コンバージェンスが無効の場合、ACL リダイレクトエントリの数は、PBR ポリシー全体の一意のプライマリ ネクストホップの数に比例します。PBR 高速コンバージェンスが有効の場合、PBR ポリシーのルートマップシーケンス全体で設定されたプライマリ ネクスト ホップとバックアップ ネクスト ホップの固有の組み合わせの数に比例する ACL リダイレクト エントリがポート スライスごとに必要になることがあります。
- 次のプラットフォームが PBR高速コンバージェンスをサポートします。N9K-C93180YC-FX、N9K-C93180YC2-FX、N9K-C93180YC-FX-24、N9K-C93108TC-FX、N9K-C93108TC2-FX、N9K-C93108TC-FX-24、N9K-C9336C-FX2、N9K-C93240YC-FX2、N9K-C93360YC-FX2、N9K-C93216TC-FX2、N9K-C9336C-FX2-E、N9K-C9316D-GX、N9K-C93600CD-GX、N9K-C9364C-GX。

- Cisco NX-OS リリース 10.3(2)F 以降、PBR のデフォルトの IPv4/IPv6 ネクストホップ VRF 選択は、Cisco Nexus 9000 シリーズ プラットフォーム スイッチで提供されます。
- Cisco NX-OS リリース 10.3(2)F 以降、IP トンネルを介した PBR は、gre および ipip モードのトンネルでのみサポートされます。ただし、IP 経由の PBR トンネルでは、 **set {ip | ipv6} next-hop** コマンドのすべてのバリアントで **load-share** キーワードがサポートされていません。
- 次のガイドラインと制約事項は、PBR セット VRF に適用されます：
  - Cisco NX-OS リリース 10.5(2)F 以降、 **set vrf** および **pbr set-vrf recirculation interface** *port-channel-num* コマンドは、Cisco Nexus 9300-FX2、FX3、GX、GX2、H2R、H1 シリーズ スイッチ、9700 GX ライン カードを搭載した 9500 シリーズ スイッチ、および N9K-C9504-FM-G または N9K-C9508-FM-G ファブリック モジュールでサポートされます。。ただし、混合モードは EoR ではサポートされていません。
  - VXLAN エンドポイント、GRE、IP-in-IP トンネルなどのトンネルに解決されるネクストホップはサポートされていません。

ポリシーベース ルーティングのデフォルト設定

表 *29*：デフォルトのポリシーベース ルーティング パラメータ

パラメータ	デフォルト
ポリシーベース ルーティング	無効化

ポリシーベース ルーティングの設定

ポリシーベース ルーティング機能のイネーブル化

ルートポリシーを設定する前に、ポリシーベースルーティング機能をイネーブルにしておく必要があります。

手順の概要

1. **configure terminal**
2. [**no**] **feature pbr**
3. （任意） **show feature**
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバルコンフィギュレーションモードを開始します。

Step 2	[**no**] **feature pbr** 例： `switch(config)# feature pbr`	ポリシーベース ルーティング機能をイネーブルにします。 ポリシーベースルーティング機能を無効にするには、このコマンドの **no** 形式を使用します。 （注） **no feature pbr** コマンドは、インターフェイスに適用されているポリシーを削除します。ACL またはルートマップ設定は削除されず、システムチェックポイントも作成されません。
Step 3	（任意） **show feature** 例： `switch(config)# show feature`	有効および無効にされた機能を表示します。
Step 4	（任意） **copy running-config startup-config** 例：	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

	コマンドまたはアクション	目的
	`switch(config)# copy running-config startup-config`	

ECMP 上のポリシーベース ルーティングの有効化

ECMP を介した PBR は、デフォルトでは有効になっていません。ルート ポリシーを設定する前に、ポリシーベース ルーティング機能をイネーブルにしておく必要があります。

手順の概要

1. **configure terminal**
2. [**no**] **feature pbr**
3. （任意） **show feature**
4. **hardware profile pbr ecmp paths** <*maxpath*> | **[no] hardware profile pbr ecmp paths** [<*maxpath*>]
5. **show system internal rpm state**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	[**no**] **feature pbr** 例： `switch(config)# feature pbr`	ポリシーベース ルーティング機能をイネーブルにします。 ポリシーベースルーティング機能を無効にするには、このコマンドの **no** 形式を使用します。 （注） **no feature pbr** コマンドは、インターフェイスに適用されているポリシーを削除します。ACL またはルートマップ設定は削除されず、システムチェックポイントも作成されません。
Step 3	（任意） **show feature** 例： `switch(config)# show feature`	有効および無効にされた機能を表示します。

	コマンドまたはアクション	目的

Step 4	**hardware profile pbr ecmp paths** <*maxpath*> | **[no] hardware profile pbr ecmp paths** [<*maxpath*>]
	例：
	`switch(config)# hardware profile pbr ecmp paths 12`
	`Warning!!: The pbr ecmp path limits have been changed.`
	`Please reload the switch now for the change to take effect.`
	`switch(config)#`
	`switch(config)# no hardware profile pbr ecmp paths 12`
	`Warning!!: The pbr ecmp path limits have been changed.`
	`Please reload the switch now for the change to take effect.`
	`switch(config)#`
	`switch(config)# no hardware profile pbr ecmp paths`
	`Warning!!: The pbr ecmp path limits have been changed.`
	`Please reload the switch now for the change to take effect.`
	`switch(config)#`
		IP ネクストホップの ECMP パスの数を設定します。ただし、設定された IP ネクストホップでロードシェアを明示的に設定しない限り、トラフィックはすべてのパスを通過しない可能性があります。PBR ECMP パスを削除または変更すると、その変更は次のリロード後にのみ有効になります。範囲は 1 ～ 64 です。

Step 5	**show system internal rpm state**	PBR ECMP パスの現在設定されている値と動作値を表示します。

PBR 高速コンバージェンスの設定

現在 PBR で使用されているネクストホップで障害が発生した場合、PBR高速コンバージェンスによってトラフィックのコンバージェンス時間が1秒未満に短縮されます。PBR高速コンバージェンスは、複数の代替ネクストホップで定義されたルートマップ シーケンスを持つポリシーを支援します。このオプションは、ロード シェアリング オプションを使用せず、ネクスト ホップの可用性を追跡するための SLA プローブを使用します。

PBR高速コンバージェンスは、スイッチではデフォルトで無効になっています。PBR高速コンバージェンスを設定し、設定を保存した後、スイッチをリロードして PBR高速コンバージェンスをアクティブにする必要があります。

始める前に

PBR 高速コンバージェンスを設定するには、まずポリシーベース ルーティング機能を有効にしておく必要があります。

手順の概要

1. **configure terminal**
2. [**no**] **feature pbr**
3. [**no**] **hardware profile pbr next-hop fast-convergence**
4. **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	[**no**] **feature pbr** 例： `switch(config)# feature pbr`	ポリシーベース ルーティング機能をイネーブルにします。
Step 3	[**no**] **hardware profile pbr next-hop fast-convergence** 例： `switch(config)# hardware profile pbr next-hop fast-convergence`	PBR高速コンバージェンスを設定します。 PBR 高速コンバージェンスを無効にするには、このコマンドの **no** 形式を使用します。 （注） PBR高速コンバージェンスのイネーブル化またはディセーブル化は、スイッチのリロード後に有効になります。
Step 4	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

例

次の例では、PBR高速コンバージェンスをイネーブルにし、スイッチをリロードします。

```
switch(config)# hardware profile pbr next-hop fast-convergence
Warning: Please save config and reload the system for the configuration to take effect.
switch(config)# copy running-config startup-config
switch(config)# reload
```

次のタスク

PBR高速コンバージェンスをイネーブルまたはディセーブルにし、設定を保存したら、スイッチをリロードします。

ルート ポリシーの設定

ポリシーベースルーティングでルートマップを使用すると、着信インターフェイスにルーティング ポリシーを割り当てることができます。Cisco NX-OS はネクスト ホップおよびインターフェイスを検出するときに、パケットをルーティングします。

手順の概要

1. **configure terminal**
2. **interface** *type slot/port*
3. **ip policy route-map** *map-name*
4. **ipv6 policy route-map** *map-name*
5. **match** {**ip** | **ipv6**} **address [accesslist-name]**
6. **set** {**ip** | **ipv6**} **next-hop** *address1* [*address2...*]**[load-share] [drop-on-fail] [force-order]**
7. **set** {**ip** | **ipv6**} **vrf** *vrf-name* **next-hop** *address1* [*address2...*]**[force-order] [drop-on-fail][load-share]**
8. **set** {**ip** | **ipv6**} **default next-hop** *address2* [*address2...*] [**load-share**]
9. **set** {**ip** | **ipv6**} **default vrf** *vrf-name* **next-hop** *address1* [*address2...*]**[load-share]**
10. **set** {**ip** | **ipv6**} **next-hop verify-availability** *next-hop-address* **track** *object*
11. **set** {**ip** | **ipv6**} **vrf** *vrf-name* **next-hop verify-availability** *next-hop-address* **track** *object*
12. **set** {**ip** | **ipv6**} **default next-hop verify-availability** *next-hop-address* **track** *object*
13. **set** {**ip** | **ipv6**} **default vrf** *vrf-name* **next-hop verify-availability** *next-hop-address* **track** *object*
14. **set interface** {*null0*}
15. **set vrf** *vrf-name*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例： `switch# configure terminal`	グローバル コンフィギュレーション モードを開始します。

Step 2	**interface** *type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**ip policy route-map** *map-name*
	例：
	`switch(config-if)# ip policy route-map Testmap`
	`switch(config-route-map)#`
		IPv4 ポリシーベース ルーティング用のルート マップをインターフェイスに割り当てます。

Step 4	**ipv6 policy route-map** *map-name* 例：	IPv6 ポリシーベース ルーティング用のルート マップをインターフェイスに割り当てます。

	コマンドまたはアクション	目的
	`switch(config-if)# ipv6 policy route-map Testmap` `switch(config-route-map)#`	

Step 5	**match** {**ip** | **ipv6**} **address [accesslist-name]**
	例：
	インターネットユーザに商品やサービスを提供する IPv4
	`switch(config-route-map)# match ip address ACL1_v4`
	IPv6 の場合
	`switch(config-route-map)# match ipv6 address ACL1_v6`
		1 つまたは複数の IPv4 または IPv6 アクセス コントロール リスト（ACL）に対して IP または IPv6 アドレスを照合します。このコマンドはポリシーベース ルーティング用であり、ルート フィルタリングまたは再配布では無視されます。

Step 6	**set** {**ip** | **ipv6**} **next-hop** *address1* [*address2...*]**[load-share] [drop-on-fail] [force-order]**
	例：
	インターネットユーザに商品やサービスを提供する IPv4
	`switch(config-route-map)# set ip next-hop 192.0.2.1`
	IPv6 の場合
	`switch(config-route-map)# set ipv6 next-hop 2001:0DB8::1`
		ポリシーベース ルーティング用の IPv4、または IPv6 ネクスト ホップ アドレスを設定します。このコマンドでは、複数のアドレスが設定されている場合に、最初の有効なネクスト ホップ アドレスが使用されます。
		任意の **load-share** キーワードを使用して、最大 32 のネクスト ホップ アドレスにトラフィックのロード バランシングを行います。
		CLI で指定されたネクストホップ順序を有効にするには、オプションの **force-order** キーワードを使用します。
		設定されたネクストホップが到達不能になったときに、デフォルトルーティングを使用する代わりに、オプションの **drop-on-fail** キーワードを使用してパケットをドロップできます。Cisco Nexus 9200、9300-EX/FX/FX2 および 9364C プラットフォーム スイッチ、および -EX および -FX ライン カードを備えた Cisco Nexus 9500 プラットフォーム スイッチがサポートされています。

Step 7	**set** {**ip** | **ipv6**} **vrf** *vrf-name* **next-hop** *address1* [*address2...*]**[force-order] [drop-on-fail][load-share]** 例： インターネットユーザに商品やサービスを提供する IPv4 `switch(config-route-map)# set ip vrf vrf1 next-hop 192.0.2.2` IPv6 の場合	ポリシーベースルーティングのデフォルトまたはユーザ定義の vrf に基づいて、IPv4 または IPv6 ネクスト ホップアドレスを設定します。 このコマンドは、VRF インターフェイスに到着する VRF間ルーティングパケットが、設定されたネクスト ホップに基づいて他の VRF を介してルーティングされることをサポートします。

	コマンドまたはアクション	目的
	`switch(config-route-map)# set ipv6 vrf vrf1 next-hop 2001:0DB8::1`	このコマンドでは、複数のアドレスが設定されている場合に、最初の有効なネクスト ホップ アドレスが使用されます。 CLI で指定されたネクストホップ順序を有効にするには、オプションの **force-order** キーワードを使用します。 設定されたネクストホップが到達不能になったときに、デフォルトルーティングを使用する代わりに、オプションの **drop-on-fail** キーワードを使用してパケットをドロップできます。Cisco Nexus 9200、9300-EX/FX/FX2 および 9364C プラットフォーム スイッチ、および -EX および -FX ライン カードを備えた Cisco Nexus 9500 プラットフォーム スイッチがサポートされています。 任意の **load-share** キーワードを使用して、最大 32 のネクスト ホップ アドレスにトラフィックのロード バランシングを行います。

Step 8	**set** {**ip** | **ipv6**} **default next-hop** *address2* [*address2...*] [**load-share**]
	
	例：
	
	インターネットユーザに商品やサービスを提供する IPv4
	
	`switch(config-route-map)#set ip default next-hop 192.0.2.2`
	
	IPv6 の場合
	
	`switch(config-route-map)#set ipv6 default next-hop 2001:0DB8::1`
		宛先への明示的ルートがない場合に使用する、ポリシーベース ルーティング用の IPv4、またはIPv4 ネクストホップアドレスを設定します。このコマンドでは、複数のアドレスが設定されている場合に、最初の有効なネクスト ホップ アドレスが使用されます。これは、ネクストホップトラッキングでのみ実行できます。
		
		• 任意の **load-share** キーワードを使用して、最大 32 のネクストホップ アドレスにトラフィックのロード バランシングを行います。
		
		Cisco NX-OS リリース 10.2(2)F 以降、以下がサポートされます。
		
		• **set ip default next-hop** コマンドは、GX、GX2、およびFX3プラットフォーム スイッチでサポートされています。
		
		• 追跡対象オブジェクトの到達可能性を確認するには、オプションの **verify-availability** キーワードを使用します。
		
		（注）
		このコマンドは N9K-C950x で現在サポートされていません。

	コマンドまたはアクション	目的

Step 9	**set** {**ip** | **ipv6**} **default vrf** *vrf-name* **next-hop** *address1* [*address2...*]**[load-share]**
	例：
	インターネットユーザに商品やサービスを提供する IPv4
	`switch(config-route-map)# set ip default vrf vrf1 next-hop 192.0.2.2`
	IPv6 の場合
	`switch(config-route-map)# set ipv6 default vrf vrf1 next-hop 2001:0DB8::1`
		宛先への明示的ルートがない場合に使用する、ポリシーベース ルーティング用の IPv4、またはIPv4 ネクストホップ アドレスを設定します。
		このコマンドは、VRFインターフェイスに到着するVRF間ルーティングパケットが、構成されたネクスト ホップに基づいて他の VRF を介してルーティングされることをサポートします。
		このコマンドでは、複数のアドレスが設定されている場合に、最初の有効なネクスト ホップ アドレスが使用されます。
		（注）
		このコマンドでは、set ステートメントで複数の VRF を使用できません。
		任意の **load-share** キーワードを使用して、最大 32 のネクスト ホップ アドレスにトラフィックのロード バランシングを行います。

Step 10	**set** {**ip** | **ipv6**} **next-hop verify-availability** *next-hop-address* **track** *object* 例： `switch(config-route-map)# set ip next-hop verify-availability 192.0.2.2 track 1`	ポリシーベースルーティング用のIPv4、またはIPv6 ネクスト ホップ アドレスを設定します。 スイッチがそのネクスト ホップへのポリシー ルーティングを実行する前に、ルートマッピングのネクスト ホップの到達可能性を確認するポリシー ルーティングを設定するには、このコマンドを使用します。この手順を繰り返して、他のトラッキング対象オブジェクトの到達可能性を確認するためのルートマップを設定します。 （注） オブジェクト トラッキングに関する詳細情報ついては、『Cisco Nexus 9000 Series NX-OS IP SLAs Configuration Guide』を参照してください。
Step 11	**set** {**ip** | **ipv6**} **vrf** *vrf-name* **next-hop verify-availability** *next-hop-address* **track** *object* 例： `switch(config-route-map)# set ip vrf vrf1 next-hop verify-availability 192.0.2.2 track 1`	ポリシーベースルーティングのデフォルトまたはユーザ定義の vrf に基づいて、IPv4 または IPv6 ネクスト ホップアドレスを設定します。 このコマンドは、VRFインターフェイスに到着するVRF間ルーティングパケットが、設定されたネクスト ホップに基づいて他の VRF を介してルーティングされることをサポートします。 スイッチがそのネクスト ホップへのポリシー ルーティングを実行する前に、ルートマッピングのネク

	コマンドまたはアクション	目的
		スト ホップの到達可能性を確認するポリシー ルーティングを設定するには、このコマンドを使用します。この手順を繰り返して、他のトラッキング対象オブジェクトの到達可能性を確認するためのルートマップを設定します。 （注） オブジェクト トラッキングの設定の詳細については、『Cisco Nexus 9000 Series NX-OS IP SLAs Configuration Guide』を参照してください。
Step 12	**set** {**ip** | **ipv6**} **default next-hop verify-availability** *next-hop-address* **track** *object* 例： `switch(config-route-map)# set ip default next-hop verify-availability 192.0.2.2 track 1`	宛先への明示的ルートがない場合に使用する、ポリシーベース ルーティング用の IPv4、またはIPv4 ネクストホップ アドレスを設定します。 スイッチがそのネクスト ホップへのポリシー ルーティングを実行する前に、ルートマッピングのネクスト ホップの到達可能性を確認するポリシー ルーティングを設定するには、このコマンドを使用します。この手順を繰り返して、他のトラッキング対象オブジェクトの到達可能性を確認するためのルートマップを設定します。 （注） オブジェクト トラッキングに関する詳細情報ついては、『Cisco Nexus 9000 Series NX-OS IP SLAs Configuration Guide』を参照してください。
Step 13	**set** {**ip** | **ipv6**} **default vrf** *vrf-name* **next-hop verify-availability** *next-hop-address* **track** *object* 例： `switch(config-route-map)# set ip default vrf vrf1 next-hop verify-availability 192.0.2.2 track 1`	宛先への明示的ルートがない場合に使用する、ポリシーベース ルーティング用の IPv4、またはIPv4 ネクストホップ アドレスを設定します。 このコマンドは、VRFインターフェイスに到着するVRF間ルーティング パケットが、構成されたネクスト ホップに基づいて他の VRF を介してルーティングされることをサポートします。 スイッチがそのデフォルト VRF ネクスト ホップへのポリシー ルーティングを実行する前に、ルートマッピングのネクストホップの到達可能性を確認するポリシー ルーティングを構成するには、このコマンドを使用します。この手順を繰り返して、他のトラッキング対象オブジェクトの到達可能性を確認するためのルート マップを設定します。 （注）

	コマンドまたはアクション	目的
		オブジェクト トラッキングの設定の詳細については、『Cisco Nexus 9000 Series NX-OS IP SLAs Configuration Guide』を参照してください。
Step 14	**set interface** {*null0*} 例： `switch(config-route-map)# set interface null0`	ルーティングに使用するインターフェイスを設定します。パケットをドロップするには **null0** インターフェイスを使用します。
Step 15	**set vrf** *vrf-name* 例： `switch(config-route-map)# set vrf vrf1`	ルートマップで **set vrf** コマンドを使用すると、ルート ルックアップまたはネクストホップ解決に使用する VRF コンテキストを指定できます。 （注） **set vrf** コマンドが機能するために必要な前提条件の設定がいくつかあります。詳細については、「ルートマップを使用した VRF へのリダイレクト（640 ページ）」の項を参照してください。

ルートマップを使用した VRF へのリダイレクト

ルートマップで **set vrf** コマンドを使用すると、ルート ルックアップまたはネクストホップ解決に使用する VRF コンテキストを指定できます。このコマンドを使用すると、入力インターフェイスに関連付けられた VRF をデフォルトにするのではなく、ルートのルックアップに使用する VRF を制御できます。

set VRF 機能は、2 パス ソリューションを通じてサポートされます。そのためには、専用のリンク ループバック インターフェイスを使用して、パケットを目的の VRF にリダイレクトします。

- パス 1 では、ルートマップ評価後に PBR インターフェイスに入るパケットが、リンク ループバック モードでメンバーポートが構成されている専用ポートチャネル dot1q インターフェイスにリダイレクトされます。ポートチャネル dot1q インターフェイスは、目的のリダイレクト VRF のメンバーである必要があります。
- パス 2 では、メンバー ポートのリンクループバック構成により、パケットは同じ専用ポートチャネル dot1q インターフェイスに再循環され、目的の VRF でデフォルトルーティングが行われます。

pbr set-vrf recirculation interface port-channel コマンドは、VRF を再循環ポート チャネル サブインターフェイス マッピングに設定するために使用されます。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **pbr set-vrf recirculation interface port-channel** *port-channel-num*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例： `switch# configure terminal`	グローバル コンフィギュレーション モードを開始します。

Step 2	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context vrf1`
	`switch(config-if)#`
		VRF 設定モードを開始します。
		（注）
		再循環インターフェイスがデフォルトの VRF 用に構成されている場合、この手順は必要ありません。

Step 3	**pbr set-vrf recirculation interface port-channel** *port-channel-num* 例： `switch(config-if)# pbr set-vrf recirculation interface port-channel 1.1`	ポートチャネルを使用してパケットを VRF にリダイレクトする VRF コンテキストで、set-vrf 再循環インターフェイスを構成します。

VRF の構成

set vrf コマンドを使用する手順は、次のとおりです：

1. グローバル構成モードを開始します。

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

2. 機能 PBR を有効にします。

```
switch(config)# feature pbr
```

3. ポートチャネル インターフェイスを作成します。

```
switch(config)# interface port-channel1
switch(config-if)# mtu 9216
switch(config-if)# no shutdown
```

4. イーサネット インターフェイスでリンクループバック設定を有効にし、チャネル グループに割り当てます。

```
switch# configure terminal
switch(config)# interface Ethernet1/1
switch(config-if)# mtu 9216
switch(config-if)# link loopback
switch(config-if)# channel-group 1
switch(config-if)# no shutdown
```

5. 目的の VRF へのパケット転送に必要な構成を使用して、ポートチャネル サブインターフェイスを作成します。

```
switch# configure terminal
switch(config)# interface port-channel1.1
switch(config-if)# encapsulation dot1q 101
switch(config-if)# vrf member vrf1
switch(config-if)# ip forward
switch(config-if)# ipv6 address use-link-local-only
switch(config-if)# ipv6 nd dad attempts 0
switch(config-if)# ipv6 nd prefix default no-advertise
switch(config-if)# ipv6 nd suppress-ra
switch(config-if)# no shutdown
```

6. VRF から再循環ポート チャネル サブインターフェイスへのマッピングを定義します。

```
switch# configure terminal
switch(config)# vrf context vrf1
switch(config-vrf)# pbr set-vrf recirculation interface port-channel1.1
```

7. **set vrf** *vrf-name* コマンドを使用してルートマップを設定します。

```
switch# configure terminal
switch(config)# route-map test permit 10
switch(config-route-map)# match ip address acl1
switch(config-route-map)# set vrf vrf1
switch(config-route-map)# route-map testv6 permit 10
switch(config-route-map)# match ip address acl1_v6
switch(config-route-map)# set vrf vrf1
```

8. 入力インターフェイスにポリシーを適用します。

```
switch# configure terminal
switch(config)# interface eth1/10
switch(config-if)# ip policy route-map test
switch(config-if)# ip policy route-map testv6
```

（注）

- set vrf 機能用に構成されたリンクループバックインターフェイス、ポートチャネルおよびポートチャネルサブインターフェイスは、他の目的に使用しないでください。
- 追加の帯域幅が必要な場合は、チャネル グループにメンバー ポートを追加できます。
- リンクループバック構成は、各メンバーポートで明示的に有効にする必要があります。
- 新しいポート チャネル サブインターフェイスは、上記の手順を使用して、同じチャネル グループ内に作成することも、さまざまな VRF リダイレクションに対して別のチャネル グループに作成することもできます。
- 各ポートチャネル再循環サブインターフェイスには、一意の dot1q ラベルが必要です。
- ポートチャネル再循環サブインターフェイスに割り当てられた VRF メンバーが、pbr set-vrf 再循環インターフェイスが設定されている VRF コンテキストと一致することを確認します。
- pbr set vrf 機能を使用している場合、2 パス ソリューションが原因でパケット遅延がわずかに増加する可能性があります。
- チャネル内のすべてのメンバー ポートが同じポート速度に構成されていることを確認します。
- IPv6 関連の構成が再循環 PO サブインターフェイスで行われると、次の syslog メッセージが生成されます。

```
2024 Jun 17 22:08:07 Leaf-switch %ICMPV6-3-ND_LOG: icmpv6 [16250] Own mac address
ecce.13e2.271f in the NA packet received on port-channel1.1 from fe80::eece:13ff:fee2:271f
2024 Jun 17 22:56:58 Leaf-switch %ICMPV6-3-ND_LOG: icmpv6 [16250] Duplicate target
address=fe80::eece:13ff:fee2:271f detected on interface=port-channel1.1 in NS packet
```

ネクストホップに一致するデフォルト ルートをリダイレクト

Cisco NX-OS リリース 10.3(3)F 以降では、デフォルト ルート一致を Cisco Nexus 9300-EX/FX/FX2/GX プラットフォーム スイッチのネクストホップにリダイレクトできます。

手順の概要

1. **configure terminal**
2. [**no**] **feature pbr**
3. **hardware access-list tcam pbr match-default-route**
4. {**ip** | **ipv6**} **policy route-map** *map-name*
5. **route-map** *map-name*
6. **match** {**ip** | **ipv6**} **address [accesslist-name]**
7. **set** {**ip** | **ipv6**} **default next-hop** *address2* [*address2...*] [**load-share**]

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	[**no**] **feature pbr** 例： `switch(config)# feature pbr`	ポリシーベース ルーティング機能をイネーブルにします。
Step 3	**hardware access-list tcam pbr match-default-route** 例： `switch(config)# hardware access-list tcam pbr match-default-route`	デフォルト ルートに一致するパケットを、ポリシー内の指定されたネクストホップにリダイレクトします。 **hardware access-list tcam pbr match-default-route** コマンドを使用すると、次の順序でトラフィックが転送されます。 特定の FIB ルート => PBR => デフォルトルートの説明：特定のルートが PBR よりも優先されます 2） （注） コマンドを有効にすると、構成されたすべての新しいポリシーで有効になります。 このコマンドが有効になっていない場合、トラフィック転送中に次の順序が実行されます。 任意の FIB ルート（特定のルートまたはデフォルト ルート）=> PBR 説明：任意のルート（特定のルートまたはデフォルト ルート）が PBR 3）よりも優先されます。

Step 4	{**ip** | **ipv6**} **policy route-map** *map-name*
	例：
	インターネット ユーザに商品やサービスを提供する IPv4
	`switch(config-if)# ip policy route-map Testmap`
	IPv6 の場合
	`switch(config-if)# ipv6 policy route-map Testmap`
		IPv4/IPv6 ポリシーベース ルーティング用のルートマップをインターフェイスに割り当てます。

	コマンドまたはアクション	目的

Step 5	**route-map** *map-name*
	例：
	`switch(config-if)# route-map Testmap`
	`switch(config-route-map)#`
		ルート マップを作成するか、または既存のルート マップに対応するルート マップ コンフィギュレーション モードを開始します。

Step 6	**match** {**ip** | **ipv6**} **address [accesslist-name]**
	例：
	インターネット ユーザに商品やサービスを提供する IPv4
	`switch(config-route-map)# match ip address ACL1_v4`
	IPv6 の場合
	`switch(config-route-map)# match ipv6 address ACL1_v6`
		1 つまたは複数の IPv4 または IPv6 アクセス コントロール リスト（ACL）に対して IP または IPv6 アドレスを照合します。このコマンドはポリシーベース ルーティング用であり、ルート フィルタリングまたは再配布では無視されます。

Step 7	**set** {**ip** | **ipv6**} **default next-hop** *address2* [*address2...*] [**load-share**]
	例：
	インターネット ユーザに商品やサービスを提供する IPv4
	`switch(config-route-map)#set ip default next-hop 192.0.2.2`
	IPv6 の場合
	`switch(config-route-map)#set ipv6 default next-hop 2001:0DB8::1`
		宛先への明示的ルートがない場合に使用する、ポリシーベース ルーティング用の IPv4、またはIPv4 ネクストホップ アドレスを設定します。このコマンドでは、複数のアドレスが設定されている場合に、最初の有効なネクスト ホップ アドレスが使用されます。これは、ネクスト ホップ トラッキングでのみ実行できます。
		• 任意の **load-share** キーワードを使用して、最大 32 のネクストホップ アドレスにトラフィックのロード バランシングを行います。
		• **set ip default next-hop** コマンドは、GX、GX2、および FX3 プラットフォーム スイッチでサポートされています。
		• 追跡対象オブジェクトの到達可能性を確認するには、オプションの **verify-availability** キーワードを使用します。

ポリシーベース ルーティングの設定の確認

ポリシーベース ルーティングの設定情報を表示するには、次のいずれかのタスクを実行します。

コマンド	目的
show [**ip** | **ipv6**] **policy** [*name*]	IPv4 または IPv6 ポリシーに関する情報を表示します。

コマンド	目的
show route-map [*name*] **pbr-statistics**	ポリシー統計情報を表示します。

ポリシー統計を有効にするには、**route-map** *map-name* **pbr-statistics** を使用します。ポリシー統計を消去するためには、**clear route-map** *map-name* **pbr-statistics**コマンドを使用します。

ポリシーベース ルーティングの設定例

インターフェイス上で単純なルート ポリシーを設定する例を示します。

```
feature pbr
ip access-list pbr-sample_1
  permit tcp host 10.1.1.1 host 192.168.2.1 eq 80
ip access-list pbr-sample_2
  permit tcp host 10.1.1.2 host 192.168.2.2 eq 80
!
route-map pbr-sample permit 10
match ip address pbr-sample_1
set ip next-hop 192.168.1.1
route-map pbr-sample permit 20
match ip address pbr-sample_2
set ip next-hop 192.168.1.2
!
route-map pbr-sample pbr-statistics

interface ethernet 1/2
  ip policy route-map pbr-sample
```

次の出力で、この設定を確認します。

```
switch# show route-map pbr-sample

route-map pbr-sample, permit, sequence 10
 Match clauses:
   ip address (access-lists): pbr-sample_1
Set clauses:
   ip next-hop 192.168.1.1
route-map pbr-sample, permit, sequence 20
 Match clauses:
   ip address (access-lists): pbr-sample_2
 Set clauses:
   ip next-hop 192.168.1.2

switch# show route-map pbr-sample pbr-statistics

route-map pbr-sample, permit, sequence 10
Policy routing matches: 84 packets

route-map pbr-sample, permit, sequence 20
Policy routing matches: 94 packets

Default routing: 233 packets
```

（注） すべてのルートマップ シーケンスに対して表示される**ポリシー ルーティング マッチ数**には、ルートマップ内のシーケンスとマッチする着信データトラフィックのパケット数が含まれます。このカウンタは、PBR リダイレクション（そのシーケンスの「set」コマンド）が解決されたかどうかに関係なく増加します。同様に、上記の例では、show route-map pbr-statistics pbr-sample の出力の 2 つのルートマップ シーケンス（シーケンス 10 と 20）に対するポリシー ルーティング マッチ数が示されています。

（注） **デフォルトルーティング**には、ルートマップ内のどのシーケンスともマッチしない着信データトラフィックのパケット数が含まれます。同様に上記の例では、デフォルトルーティングは、show route-map pbr-statistics pbr-sample 出力の最後に 1 回だけ表示されます。

この例は、ECMP パスと非 ECMP パス間のロード シェアリングを示しています。

```
switch# show run rpm
!Command: show running-config rpm
!Running configuration last done at: Sun Dec 23 16:02:32 2018
!Time: Sun Dec 23 16:06:13 2018

version 9.2(3) Bios:version 08.35
feature pbr

route-map policy1 pbr-statistics
route-map policy1 permit 10
  match ip address acl2
  set ip next-hop 131.1.1.2 load-share
route-map policy2 pbr-statistics
route-map policy2 permit 10
  match ip address acl2
  set ip next-hop verify-availability 131.1.1.2 track 1
  set ip next-hop verify-availability 30.1.1.2 track 2 load-share


interface Ethernet1/31
  ip policy route-map policy2
```

この例は、ネクスト ホップ ルーティング要求に関する情報を表示しています。

```
switch# show system internal rpm pbr ip nexthop
PBR IPv4 nexthop table for vrf default

30.1.1.2 Usable
  via 28.1.1.2 Ethernet1/18 a46c.2ae3.02a7

131.1.1.2 Usable
  via 111.1.1.2 Vlan81 8478.ac58.afc1
Usable
  via 112.1.1.2 Vlan82 8478.ac58.afc1
Usable
  via 113.1.1.2 Vlan83 8478.ac58.afc1
Usable
  via 114.1.1.2 Vlan84 8478.ac58.afc1
Usable
```

```
  via 115.1.1.2 Vlan85 8478.ac58.afc1
Usable
  via 116.1.1.2 Vlan86 8478.ac58.afc1
Usable
  via 117.1.1.2 Vlan87 8478.ac58.afc1
Usable
  via 118.1.1.2 Vlan88 8478.ac58.afc1
```

この例は、ユニキャスト RIB から受け取ったルートを表示しています。

```
switch# show ip route 130.1.1.2
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>

130.1.1.0/24, ubest/mbest: 8/0
    *via 111.1.1.2, Vlan81, [110/120], 00:07:57, ospf-1, inter
    *via 112.1.1.2, Vlan82, [110/120], 00:07:57, ospf-1, inter
    *via 113.1.1.2, Vlan83, [110/120], 00:07:57, ospf-1, inter
    *via 114.1.1.2, Vlan84, [110/120], 00:07:57, ospf-1, inter
    *via 115.1.1.2, Vlan85, [110/120], 00:07:57, ospf-1, inter
    *via 116.1.1.2, Vlan86, [110/120], 00:07:57, ospf-1, inter
    *via 117.1.1.2, Vlan87, [110/120], 00:07:57, ospf-1, inter
    *via 118.1.1.2, Vlan88, [110/120], 00:07:57, ospf-1, inter


switch# show ip route 30.1.1.2
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>

30.1.1.0/24, ubest/mbest: 1/0
    *via 28.1.1.2, [1/0], 00:38:36, static
```

次に、vrfベースのネクストホップを使用したポリシーベースルーティングの例を示します。

```
route-map policy_vrf_default_v4 permit 10
  match ip address acl1_v4_tc1
  set ip vrf default next-hop 31.1.1.1

route-map policy_vrf_nondefault_v4 permit 10
  match ip address acl1_v4_tc2
  set ip vrf vrf1 next-hop 32.1.1.1

show route-map policy_vrf_default_v4
route-map policy_vrf_default_v4, permit, sequence 10
  Match clauses:
    ip address (access-lists): acl1_v4_tc1
  Set clauses:
    ip vrf default next-hop 31.1.1.1

show route-map policy_vrf_nondefault_v4
route-map policy_vrf_nondefault_v4, permit, sequence 10
  Match clauses:
    ip address (access-lists): acl1_v4_tc2
  Set clauses:
    ip vrf vrf1 next-hop 32.1.1.1
```

次の例では、デフォルトのネクストホップを使用したポリシーベースルーティングを示します。

```
route-map policy_default_v4 permit 10
  match ip address acl1_v4_tc1
  set ip default next-hop 21.1.1.2
```

```
show route-map policy_default_v4
route-map policy_default_v4, permit, sequence 10
  Match clauses:
    ip address (access-lists): acl1_v4_tc1
  Set clauses:
    ip default next-hop 21.1.1.2
```

次に、vrf ベースのデフォルト ネクスト ホップを使用したポリシーベース ルーティングの例を示します。

```
route-map policy_default_vrf_default_v4 permit 10
  match ip address acl1_v4_tc1
  set ip default vrf default next-hop 21.1.1.2
route-map policy_default_vrf_nondefault_v4 permit 10
  match ip address acl1_v4_tc1
  set ip default vrf vrf1 next-hop 22.1.1.2
```

```
show route-map policy_default_vrf_default_v4
route-map policy_default_vrf_default_v4, permit, sequence 10
  Match clauses:
    ip address (access-lists): acl1_v4_tc1
  Set clauses:
    ip default vrf default next-hop 21.1.1.2
show route-map policy_default_vrf_nondefault_v4
route-map policy_default_vrf_nondefault_v4, permit, sequence 10
  Match clauses:
    ip address (access-lists): acl1_v4_tc1
  Set clauses:
    ip default vrf vrf1 next-hop 22.1.1.2
```

ポリシーベースルーティングの関連資料

関連項目	マニュアル タイトル
IP SLA PBR オブジェクト トラッキング	『Cisco Nexus 9000 Series NX-OS IP SLAs Configuration Guide』
トラブルシューティング情報	『Cisco Nexus 9000 Series NX-OS Troubleshooting Guide』

第 19 章

『Configuring HSRP』

この章は、次の項で構成されています。

- HSRP について （651 ページ）
- HSRP サブネット VIP （656 ページ）
- HSRP 認証 （656 ページ）
- HSRP メッセージ （656 ページ）
- HSRP ロード シェアリング （657 ページ）
- オブジェクト トラッキングおよび HSRP （658 ページ）
- vPC と HSRP （658 ページ）
- BFD （658 ページ）
- ハイ アベイラビリティおよび拡張ノンストップ フォワーディング （659 ページ）
- 仮想化のサポート （659 ページ）
- HSRP の前提条件 （659 ページ）
- HSRP の注意事項と制約事項 （659 ページ）
- HSRP パラメータのデフォルト設定 （662 ページ）
- 『Configuring HSRP』 （662 ページ）
- HSRP 設定の確認 （678 ページ）
- HSRP の設定例 （679 ページ）
- その他の参考資料 （680 ページ）

HSRP について

HSRP はファーストホップ冗長プロトコル（FHRP）であり、ファーストホップ IP ルータの透過的なフェールオーバーを可能にします。HSRP は、デフォルト ルータの IP アドレスを指定して設定された、イーサネット ネットワーク上の IP ホストにファーストホップ ルーティングの冗長性を提供します。ルータ グループでは HSRP を使用して、アクティブ ルータおよびスタンバイ ルータを選択します。ルータ グループでは、アクティブルータはパケットをルーティングするルータです。スタンバイルータは、アクティブルータで障害が発生した場合、または事前に設定された条件が満たされた場合に、引き継ぐルータです。

大部分のホストの実装では、ダイナミックなルータディスカバリ メカニズムをサポートしていませんが、デフォルトのルータを設定することはできます。すべてのホスト上でダイナミックなルータ ディスカバリ メカニズムを実行するのは、管理上のオーバーヘッド、処理上のオーバーヘッド、セキュリティ上の問題など、さまざまな理由で現実的ではありません。HSRP は、そうしたホスト上にフェールオーバー サービスを提供します。

HSRP の概要

HSRP を使用する場合、HSRP の仮想 *IP* アドレスを（実際のルータの IP アドレスではなく）ホストのデフォルト ルータとして設定します。仮想 IP アドレスは、HSRP が動作するルータのグループで共有される IPv4 または IPv6 アドレスです。

ネットワーク セグメントに HSRP を設定する場合は、HSRP グループ用の仮想 *MAC* アドレスと仮想 IP アドレスを設定します。グループの各 HSRP 対応インターフェイス上で、同じ仮想アドレスを指定します。各インターフェイス上で、実アドレスとして機能する固有の IP アドレスおよび MAC アドレスも設定します。HSRP はこれらのインターフェイスの 1 つをアクティブ ルータとして選択します。アクティブ ルータは、グループの仮想 MAC アドレス宛てのパケットを受信してルーティングします。

指定されたアクティブ ルータで障害が発生すると、HSRP によって検出されます。その時点で、選択されたスタンバイ ルータが HSRP グループの MAC アドレスおよび IP アドレスの制御を行うことになります。HSRP はこの時点で、新しいスタンバイ ルータの選択も行います。

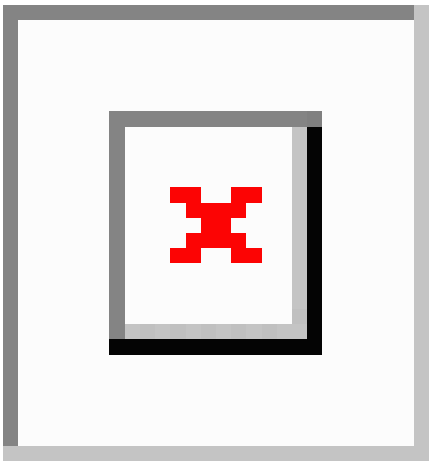
HSRP ではプライオリティ指示子を使用して、デフォルトのアクティブ ルータにする HSRP 設定インターフェイスを決定します。アクティブ ルータとしてインターフェイスを設定するには、グループ内の他のすべての HSRP 設定インターフェイスよりも高いプライオリティを与えます。デフォルトのプライオリティは 100 なので、それよりもプライオリティが高いインターフェイスを 1 つ設定すると、そのインターフェイスがデフォルトのアクティブ ルータになります。

HSRP が動作するインターフェイスは、マルチキャスト ユーザ データグラム プロトコル（UDP）ベースの hello メッセージを送受信して、障害を検出し、アクティブおよびスタンバイ ルータを指定します。アクティブ ルータが設定された時間内に hello メッセージを送信できなかった場合は、最高のプライオリティのスタンバイ ルータがアクティブ ルータになります。アクティブ ルータとスタンバイルータ間のパケット フォワーディング機能の移動は、ネットワーク上のすべてのホストに対して完全に透過的です。

1 つのインターフェイス上で複数の HSRP グループを設定できます。

次の図に、HSRP 用に設定されたネットワークのセグメントを示します。仮想 MAC アドレスおよび仮想 IP アドレスの共有によって、2 つ以上のインターフェイスが単一の仮想ルータのように動作できます。

図 *44 : 2* 台の対応ルータを含む *HSRP* トポロジ



仮想ルータは物理的には存在しませんが、相互にバックアップするように設定されたインターフェイスにとって、共通のデフォルト ルータになります。アクティブ ルータの IP アドレスを使用して、LAN 上でホストを設定する必要はありません。代わりに、仮想ルータの IP アドレス（仮想 IPアドレス）をホストのデフォルトルータとして設定します。アクティブルータが設定時間内に hello メッセージを送信できなかった場合は、スタンバイ ルータが引き継いで仮想アドレスに応答し、アクティブルータになってアクティブルータの役割を引き受けます。ホストの観点からは、仮想ルータは同じままです。

（注） ルーテッド ポートで受信した HSRP 仮想 IP アドレス宛のパケットは、ローカル ルータ上で終端します。そのルータがアクティブ HSRP ルータであるのかスタンバイ HSRP ルータであるのかは関係ありません。このプロセスには ping トラフィックと Telnet トラフィックが含まれます。レイヤ 2（VLAN）インターフェイスで受信した HSRP 仮想 IP アドレス宛のパケットは、アクティブルータ上で終端します。

HSRP のバージョン

Cisco NX-OS は、デフォルトで HSRP バージョン 1 をサポートします。HSRP バージョン 2 を使用するようにインターフェイスを設定できます。

HSRP バージョン 2 では、HSRP バージョン 1 から次のように拡張されています。

グループ番号の範囲が拡大されました。HSRP バージョン 1 がサポートするグループ番号は 0 ～ 255 です。HSRP バージョン 2 がサポートするグループ番号は 0 ～ 4095 です。

IPv4 では、HSRP バージョン 1 で使用する IP マルチキャスト アドレス 224.0.0.2 の代わりに、IPv4 マルチキャスト アドレス 224.0.0.102 または IPv6 マルチキャスト アドレス FF02::66 を使用して hello パケットを送信します。

IPv4 では 0000.0C9F.F000 ～ 0000.0C9F.FFFF、IPv6 アドレスでは 0005.73A0.0000 ～ 0005.73A0.0FFF の MAC アドレス範囲を使用します。HSRP バージョン 1 で使用する MAC アドレス範囲は、0000.0C07.AC00 ～ 0000.0C07.ACFF です。

MD 5 認証のサポートが追加されました。

HSRP のバージョンを変更すると、Cisco NX-OS がグループを再初期化します。新しい仮想 MAC アドレスがグループに与えられるからです。

HSRP バージョン 2 では HSRP バージョン 1 とは異なるパケット フォーマットを使用します。パケット フォーマットは Type-Length-Value（TLV）です。HSRP バージョン 1 ルータは、HSRP バージョン 2 パケットを受信しても無視します。

HSRP for IPv4

HSRP ルータは、HSRP hello パケットを交換することによって相互に通信します。これらのパケットは、UDP ポート 1985 上の宛先 IP マルチキャスト アドレス 224.0.0.2（すべてのルータと通信するための予約済みマルチキャストアドレス）に送信されます。アクティブルータが設定済みの IP アドレスと HSRP 仮想 MAC アドレスから hello パケットを取得するのに対して、スタンバイ ルータは、設定済みの IP アドレスとインターフェイス MAC アドレス（バーンドインアドレス（BIA）である可能性があります）から hello パケットを取得します。BIA は、MAC アドレスの下位 6 バイトで、ネットワーク カード（NIC）の製造元によって割り当てられます。

ホストはデフォルト ルータが HSRP 仮想 IP アドレスとして設定されているので、HSRP 仮想 IP アドレスに関連付けられた MAC アドレスと通信する必要があります。この MAC アドレスは、仮想 MAC アドレス 0000.0C07.ACxy です。この場合、xy はそれぞれのインターフェイスに基づく、16 進数の HSRP グループ番号です。たとえば、HSRP グループ 1 は 0000.0C07.AC01 という HSRP 仮想 MAC アドレスを使用します。隣接 LAN セグメント上のホストは、標準のアドレス解決プロトコル（ARP）プロセスを使用して、関連付けられた MAC アドレスを解決します。

HSRP バージョン 2 では新しい IP マルチキャスト アドレス 224.0.0.102 を使用して hello パケットを送信します。バージョン 1 では、このマルチキャスト アドレスが 224.0.0.2 です。バージョン 2 では、拡張グループ番号範囲 0 ～ 4095 を使用できます。また、新しい MAC アドレス範囲 0000.0C9F.F000 ～ 0000.0C9F.FFFF を使用します。

HSRP for IPv6。

IPv6 ホストは、IPv6 ネイバー探索（ND）ルータ アドバタイズメント（RA）メッセージを通じて使用可能な IPv6 ルータを学習します。これらのメッセージは、定期的にマルチキャストされる他、ホストによって送信要求されることもあります。ただし、デフォルト ルートがダウンしていることを検出したときの遅延時間は 30 秒以上になることもあります。IPv6 の HSRP は、IPv6 ND プロトコルを使用した場合よりも、代替デフォルト ルータへのスイッチオーバーが大幅に高速であり、ミリ秒タイマーが使用される場合は 1 秒未満になります。IPv6 の HSRP では、IPv6 ホストの仮想ファースト ホップを提供します。

HSRP の IPv6 インターフェイスを設定すると、IPv6 ND がルータのライフタイムがゼロで最終 RA を送信した後で、インターフェイスのリンクローカル アドレスに対する定期 RA が停止します。

インターフェイスのIPv6リンクローカルアドレスに制限はありません。他のプロトコルは、このアドレスへのパケットを送受信し続けます。

IPv6 ND は、HSRP グループがアクティブなときに、HSRP 仮想 IPv6 リンクローカル アドレスの定期 RA を送信します。これらの RA は、HSRP グループがアクティブ状態のままのときに、ルータのライフタイムがゼロで最終 RA が送信されると停止します。HSRP は、アクティブ HSRP グループ メッセージ（hello、coup、resign）でのみ仮想 MAC アドレスを使用します。

IPv6 の HSRP は、次のパラメータを使用します。

- HSRP バージョン 2
- UDP ポート 2029
- 0005.73A0.0000 ～ 0005.73A0.0FFF の範囲の仮想 MAC アドレス
- マルチキャスト リンクローカル IP 宛先アドレス FF02::66
- ホップ リミット 255

IPv6 アドレスの HSRP

HSRP IPv6 グループには、HSRP グループ番号から導出される仮想 MAC アドレス、および HSRP 仮想 MAC アドレスからデフォルトで導出される仮想 IPv6 リンクローカル アドレスがあります。仮想 IPv6 リンクローカルアドレスを形成するために HSRP IPv6 グループのデフォルトの仮想 MAC アドレスが常に使用されます。グループによって実際に使用されている仮想 MAC アドレスは関係ありません。

次の表に、ここまで説明してきたに、IPv6 ネイバー探索パケットと HSRP パケットに使用される MAC アドレスと IP アドレスを示します。

表 *30 : HSRP* および *IPv6 ND* アドレス

パケット	送信元 MAC アドレス	送信元 IPv6 アドレス	宛先 IPv6 アドレス	リンク層アドレス オプション
ネイバー送信要求（NS）	インターフェイス MAC アドレス	インターフェイス IPv6 アドレス	—	インターフェイス MAC アドレス
ルータ送信要求（RS）	インターフェイス MAC アドレス	インターフェイス IPv6 アドレス	—	インターフェイス MAC アドレス
ネイバーアドバタイズメント（NA）	インターフェイス MAC アドレス	インターフェイス IPv6 アドレス	仮想 IPv6 アドレス	HSRP 仮想 MAC アドレス
ルートアドバタイズメント（RA）	インターフェイス MAC アドレス	仮想 IPv6 アドレス	—	HSRP 仮想 MAC アドレス
HSRP（非アクティブ）	インターフェイス MAC アドレス	インターフェイス IPv6 アドレス	—	—

パケット	送信元 MAC アドレス	送信元 IPv6 アドレス	宛先 IPv6 アドレス	リンク層アドレスオプション
HSRP（アクティブ）	仮想 MAC アドレス	インターフェイス IPv6 アドレス	—	—

HSRP は、IPv6 リンクローカル アドレスをユニキャスト ルーティング情報ベース（URIB）に追加しません。リンクローカル アドレスには、セカンダリ仮想 IP アドレスがありません。

グローバル ユニキャスト アドレスの場合は、HSRP は URIB および IPv6 に仮想 IPv6 アドレスを追加します。

HSRP サブネット VIP

インターフェイス IP アドレスとは異なるサブネットに HSRP サブネット仮想 IP（VIP）アドレスを設定できます。

（注） 9636C-R、9636C-RX、および 9636Q-R ライン カードを使用して、Cisco Nexus 9508 プラットフォーム スイッチの HSRP サブネット VIP を設定できます。

この機能を使用すると、パブリック IP アドレスとして VIP を使用し、プライベート IP アドレスとしてインターフェイス IP を使用して、パブリック IPv4 アドレスを節約できます。IPv6 アドレスには、より大きな IPv6 アドレス プールが使用可能であり、ルーティング可能な IPv6 アドレスを SVI で設定して通常の HSRP で使用できるため、IPv6 アドレスには HSRP サブネット VIP は必要ありません。

また、この機能により、vPC ピアへの定期的な ARP 同期が可能になり、VIP サブネット内のホストに対して HSRP サブネット VIP が設定されている場合に、ARP が VIP をソースとして使用できるようになります。

詳細については、「HSRP の注意事項と制限事項」および「HSRP の設定例」を参照してください。

HSRP 認証

HSRP のメッセージ ダイジェスト 5（MD5）アルゴリズム認証は、HSRP スプーフィング ソフトウェアから保護し、業界標準の MD5 アルゴリズムを使用して信頼性とセキュリティを向上させています。HSRP では、認証 TLV に IPv4 または IPv6 アドレスが含まれます。

HSRP メッセージ

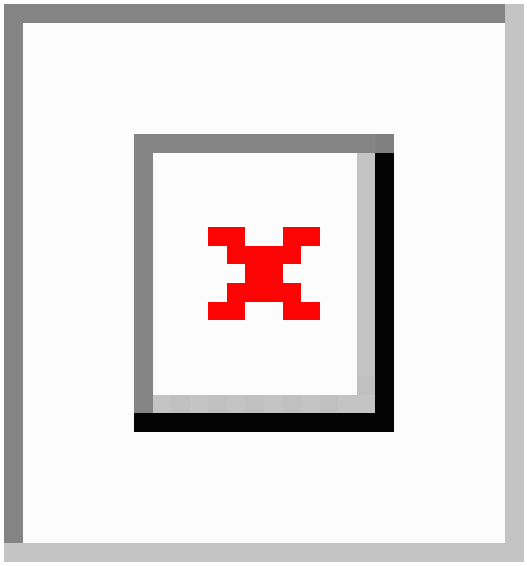
HSRP が設定されているルータは、次の 3 種類のマルチキャスト メッセージを交換できます。

- hello：hello メッセージは、ルータの HSRP プライオリティおよびステート情報を他の HSRP ルータに伝えます。
- coup：スタンバイ ルータがアクティブ ルータの機能を引き受けるときに、coup メッセージを送信します。
- resign：アクティブ ルータは、アクティブ ルータとして機能する必要がなくなったときに、このメッセージを送信します。

HSRP ロード シェアリング

HSRP では、1 つのインターフェイスに複数のグループを設定できます。オーバーラップする 2 つの IPv4 HSRP グループを設定すると、期待されるデフォルト ルータの冗長性を HSRP から提供しながら、接続ホストからのトラフィックのロード シェアリングが可能です。次の図に、ロード シェアリングが行われる HSRP IPv4 構成の例を示します。

図 *45 : HSRP* ロード シェアリング



図には、2 台のルータ（A および B）と 2 つの HSRP グループが示されています。ルータAはグループAのアクティブルータですが、グループBのスタンバイルータです。同様に、ルータBはグループBのアクティブルータであり、グループAのスタンバイルータです。両方のルータがアクティブのままの場合、HSRPは両方のルータにまたがるホスト。どちらかのルータで障害が発生すると、残りのルータが引き続き、両方のホストのトラフィックを処理します。

（注） IPv6 の HSRP では、デフォルトでロード バランシングを行います。サブネット上に 2 つの HSRP IPv6 グループが存在する場合、ホストはそれぞれのルータ アドバタイズメントから両方のグループを学習し、アドバタイズされたルータ間で負荷が共有されるように 1 つのグループを使用することを選択します。

オブジェクト トラッキングおよび HSRP

オブジェクト トラッキングを使用すると、別のインターフェイスの動作状態に基づいて、HSRP インターフェイスのプライオリティを変更できます。オブジェクト トラッキングによって、メイン ネットワークへのインターフェイスで障害が発生した場合に、スタンバイ ルータにルーティングできます。

トラッキング可能なオブジェクトは、インターフェイスのライン プロトコル ステートまたは IP ルートの到達可能性の 2 種類です。指定したオブジェクトがダウンすると、設定された値だけ Cisco NX-OS が HSRP プライオリティを引き下げます。詳細については、「HSRP オブジェクト トラッキングの設定」の項を参照してください。

vPC と HSRP

HSRP は仮想ポート チャネル（vPC）と相互運用できます。vPC を使用すると、2 個の異なる Cisco Nexus 9000 シリーズ スイッチを物理的に接続し、第 3 のデバイスからは 1 つのポートとして見えるリンクが実現します。vPC の詳細については、『Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide』を参照してください。

vPC は、アクティブ HSRP ルータとスタンバイ HSRP ルータの両方を通じてトラフィックを転送します。詳細については、「HSRP プライオリティの設定」セクションおよび「HSRP の設定例」セクションを参照してください。

（注） HSRP アクティブは、異なる SVI のプライマリおよびセカンダリ vPC ピアの両方に分散できます。

vPC ピア ゲートウェイと HSRP

一部のサード パーティ製デバイスは HSRP 仮想 MAC アドレスを無視し、代わりに HSRP ルータの送信元 MAC アドレスを使用する場合があります。vPC 環境では、この送信元 MAC アドレスを使用するパケットが vPC ピア リンク経由で送信され、それによってパケットのドロップが発生する可能性があります。vPC ピア ゲートウェイを設定して、HSRP ルータで、ローカル vPC ピア MAC アドレスとリモート vPC ピア MAC アドレス、および HSRP 仮想 MAC アドレスに送信されたパケットを直接処理できるようにします。vPC ピア ゲートウェイの詳細については、『Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide』を参照してください。

BFD

この機能では、双方向フォワーディング検出（BFD）をサポートします。BFD は、高速転送とパス障害の検出時間を提供する検出プロトコルです。BFD は 2 台の隣接デバイス間のサブセカンド障害を検出し、BFD の負荷の一部を、サポートされるモジュール上のデータ プレーンに分散でき

るため、プロトコル hello メッセージよりも CPU を使いません。詳細については、『Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide』を参照してください。

ハイアベイラビリティおよび拡張ノンストップフォワーディング

HSRP は、ステートフル リスタートおよびステートフル スイッチオーバーをサポートします。ステートフル リスタートは、HSRP プロセスが失敗してリスタートするときに行われます。ステートフル スイッチオーバーは、アクティブ スーパーバイザがスタンバイ スーパーバイザに切り替わるときに行われます。Cisco NX-OS は、スイッチオーバー後に実行コンフィギュレーションを適用します。

HSRP ホールド タイマーが短時間に設定されている場合は、これらのタイマーが切れる可能性があります。HSRP は、拡張型ノンストップ フォワーディング（NSF）をサポートし、制御されたスイッチオーバー時にこれらの HSRP ホールド タイマーを一時的に拡張します。

拡張 NSF を設定している場合、HSRP は延長されたタイマーを使用して hello メッセージを送信します。HSRP ピアは、この新しい値でホールド タイマーを更新します。タイマーが延長されることにより、スイッチオーバー中に不要な HSRP 状態の変更が発生することを防ぎます。スイッチオーバー後に、HSRP はホールド タイマーを元の設定値に復元します。スイッチオーバーに失敗すると、延長されたホールド タイマー値が満了してから HSRP はホールド タイマーを復元します。

詳細については、「HSRP の拡張ホールド タイマーの設定」の項を参照してください。

仮想化のサポート

HSRP は、仮想ルーティングおよび転送（VRF）インスタンスをサポートします。

HSRP の前提条件

- HSRP グループを設定してイネーブルにするには、その前に HSRP 機能をデバイスでイネーブルにする必要があります。

HSRP の注意事項と制約事項

HSRP 設定時の注意事項および制約事項は、次のとおりです。

- HSRP はアクティブにする前に、HSRP を設定するインターフェイスに IP アドレスを設定し、そのインターフェイスをイネーブルにします。

- 最大ホスト ルーティング モードで動作する Cisco Nexus 9500 プラットフォーム スイッチは、4 ウェイ HSRP をサポートしません。
- HSRP に IPv6 インターフェイスを設定するときは、HSRP バージョン 2 を設定する必要があります。
- IPv4 では、仮想 IP アドレスは、インターフェイス IP アドレスと同じサブネットになければなりません。
- 同一インターフェイス上では、複数のファーストホップ冗長プロトコルを設定しないことを推奨します。
- HSRP バージョン 2 は HSRP バージョン 1 と相互運用できません。どちらのバージョンも相互に排他的なので、インターフェイスはバージョン 1 およびバージョン 2 の両方を運用できません。しかし、同一ルータの異なる物理インターフェイス上であれば、異なるバージョンを実行できます。
- バージョン 1 で認められるグループ番号範囲（0 ～ 255）を超えるグループを設定している場合は、バージョン 2 からバージョン 1 へ変更することはできません。
- IPv4 に対する HSRP は、BFD でサポートされます。IPv6 に対する HSRP は、BFD でサポートされていません。
- HSRP IPv4 と HSRP IPv6 が同じ SVI の仮想 MAC アドレスを使用する場合、HSRP の状態は HSRP IPv4 と HSRP IPv6 の両方で同じである必要があります。フェールオーバー後に同じ状態になるようにするには、プライオリティとプリエンプションを設定する必要があります。
- Cisco NX-OS では、VDC、インターフェイス VRF メンバーシップ、ポート チャネル メンバーシップを変更したり、ポート モードをレイヤ 2 に変更した場合は、インターフェイス上のすべてのレイヤ 3 設定が削除されます。
- vPC で仮想 MAC アドレスを設定するときは、vPC ピアの両方で同じ仮想 MAC アドレスを設定する必要があります。
- vPC メンバである VLAN インターフェイスで HSRP MAC アドレスのバーンドイン オプションは使用できません。
- Release 7.0(3)I2(1) 以降、Cisco NX-OS ではダブルサイド vPC のすべてのノードで同じ HSRP グループを設定できます。
- 認証を設定していない場合、**show hsrp** コマンドは次の文字列を表示します。

  ```
  Authentication text "cisco"
  ```

  HSRP のデフォルトの動作は RFC 2281 で定義されています。

  ```
  If no authentication data is configured, the RECOMMENDED default
  value is 0x63 0x69 0x73 0x63 0x6F 0x00 0x00 0x00.
  ```

- 2 ペアの vPC スイッチを使用して 4 ウェイ HSRP を構成する場合（新規導入または移行シナリオ）、Nexus 9000 スイッチの vPC ペアがアクティブ/スタンバイ状態およびリッスン/リッスン状態になるように HSRP プライオリティを構成する必要があります。HSRP アクティブ/

リッスン状態、またはスタンバイ/リッスン状態にすることは Cisco Nexus 9000 vPC ピアでサポートされていません。

- この機能には、次の注意事項と制約事項があります。
  - この機能は、Cisco Nexus 9000 シリーズスイッチ、および 9636C-R、9636C-RX、および 9636Q-R ラインカードを搭載した Cisco Nexus 9508 スイッチでサポートされます。
  - この機能は、IPv4 アドレスおよび vPC トポロジでのみサポートされます。
  - プライマリまたはセカンダリ VIP をサブネット VIP にすることはできますが、サブネット VIP がインターフェイス サブネットと重複していてはなりません。
  - 通常のホスト VIP は 0 または 32 のマスク長を使用します。サブネット VIP のマスク長を指定する場合は、0 より大きく、32 未満にする必要があります。
  - URPF はこの機能ではサポートされていません。
  - VIP を使用した DHCP ソースもサポートされていません。
  - この機能では、DHCP リレーエージェントを使用して、VIP を送信元として DHCP パケットをリレーすることはできません。
  - VIP 直接ルートは、redistribute コマンドとルート マップを使用して、ルーティング プロトコルに明示的にアドバタイズする必要があります。
  - スーパーバイザが生成したトラフィック（ping、トレースルートなど）は、VIP サブネットではなく、SVI IP アドレスを使用して送信されます。
  - サブネットVIPの長さが/ 32で設定されている場合は、/ 32を指定して **no** コマンドを使用し、IPアドレスを削除する必要があります（例えば **no ip ip-address/32**、たとえば、）。
- コンフィギュレーション プロファイルを使用して設定されたサブ設定を含む SVI 設定を削除するには、まず **no interface vlan** コマンドを実行する前に、そのプロファイルを削除するか、VLAN の手動設定をクリアする必要があります。
- 次に、プリエンプション リロードタイマーを適用するための設定ガイドラインを示します。ガイドラインは、優先度の高い順にリストされています。
  1. トライアングル トポロジでは、HSRP ピアを単一の VPC ドメイン内に設定することを推奨します。この設定により、Cisco Nexus 9000 の設定がリロードされたときも、HSRP ピアでスパニングツリー ルート ブリッジが変更されなくなります。
  2. すべての VLAN のスパニング ツリー ルート ブリッジが、リロードされる Cisco Nexus 9000 上にないことを確認します。
  3. 1 と 2 が不可能な場合には、HSRP ピアではない別のスイッチに接続されているすべての SVI VLAN に対して、スイッチに有効なリンクがあることを確認します。

HSRP パラメータのデフォルト設定

デフォルトの **HSRP** パラメータ

パラメータ	デフォルト
HSRP	ディセーブル
認証	バージョン 1 の場合はテキストとしてイネーブル、パスワードは cisco
HSRP バージョン	バージョン 1
プリエンプション	ディセーブル
プライオリティ	100
仮想 MAC アドレス	HSRP グループ番号から生成

『Configuring HSRP』

HSRP の有効化

HSRP グループを設定してイネーブルにするには、その前に HSRP をグローバルでイネーブルにする必要があります。

手順の概要

1. [**no**] **feature hsrp**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	[**no**] **feature hsrp** 例： `switch(config)# feature hsrp`	HSRP 機能を有効にします。HSRP をディセーブルにするには、このコマンドの **no** 形式を使用します。

HSRP バージョン設定

HSRP のバージョンを設定できます。既存グループのバージョンを変更すると、仮想 MAC アドレスが変更されるので、Cisco NX-OS がそれらのグループの HSRP を再初期化します。HSRP のバージョンは、インターフェイス上のすべてのグループに適用されます。

（注） IPv6 HSRP グループは、HSRP バージョン 2 として設定する必要があります。

手順の概要

1. **hsrp version {1 | 2}**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**hsrp version {1 | 2}** 例： `switch(config-if)# hsrp version 2`	HSRP のバージョンを確認します。デフォルトはバージョン 1 です。

IPv4 の HSRP グループの設定

IPv4 インターフェイスに HSRP グループを設定し、その HSRP グループに仮想 IP アドレスと仮想 MAC アドレスを設定できます。

始める前に

HSRP 機能が有効になっていることを確認します（HSRP の有効化の項を参照してください）。

Cisco NX-OS では、仮想 IP アドレスを設定すると HSRP グループが有効になります。HSRP グループをイネーブルにする前に、認証、タイマー、プライオリティなどの HSRP 属性を設定する必要があります。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ip** *ip-address/length*
4. **hsrp** *group-number* [**ipv4**]
5. **ip** [*ip-address* [**secondary**]]
6. **exit**
7. **no shutdown**

8. （任意） **show hsrp** [**group** *group-number*] [**ipv4**]
9. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**ip** *ip-address/length* 例： `switch(config-if)# ip 192.0.2.2/8`	インターフェイスの IPv4 アドレスを設定します。

Step 4	**hsrp** *group-number* [**ipv4**]
	例：
	`switch(config-if)# hsrp 2`
	`switch(config-if-hsrp)#`
		HSRP グループを作成し、HSRP設定モードを開始します。HSRP バージョン 1 で指定できる範囲は 0 ～ 255 です。HSRP バージョン 2 で指定できる範囲は 0 ～ 4095 です。デフォルト値は 0 です

Step 5	**ip** [*ip-address* [**secondary**]] 例： `switch(config-if-hsrp)# ip 192.0.2.1`	HSRP グループの仮想 IP アドレスを設定し、グループを有効にします。このアドレスは、インターフェイスの IPv4 アドレスと同じサブネットになければなりません。
Step 6	**exit** 例： `switch(config-if-hsrp)# exit`	HSRP設定モードを終了します。
Step 7	**no shutdown** 例： `switch(config-if-hsrp)# no shutdown`	インターフェイスをイネーブルにします。
Step 8	（任意） **show hsrp** [**group** *group-number*] [**ipv4**] 例： `switch(config-if-hsrp)# show hsrp group 2`	HSRP 情報を表示します。

	コマンドまたはアクション	目的
Step 9	（任意） **copy running-config startup-config** 例： `switch(config-if-hsrp)# copy running-config startup-config`	実行設定を、スタートアップ設定にコピーします。

例

（注） 設定完了後にインターフェイスを有効にするには、**no shutdown** コマンドを使用する必要があります。

次に Ethernet 1/2 上で HSRP グループを設定する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
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
```

IPv6 の HSRP グループの設定

IPv6 インターフェイス上で HSRP グループを設定し、その HSRP グループに仮想 MAC アドレスを設定できます。

IPv6 の HSRP グループを設定すると、HSRP はリンクローカル プレフィックスからリンクローカル アドレスを生成します。HSRP では、Modified EUI-64 形式のインターフェイス ID も生成します。EUI-64 インターフェイス ID は、関連の HSRP 仮想 MAC アドレスから作成されます。

始める前に

HSRP は有効にする必要があります（「HSRP をイネーブルにする」のセクションを参照してください）。

IPv6 HSRP グループを設定するインターフェイスで HSRP バージョン 2 が有効になっていることを確認します。

HSRP グループをイネーブルにする前に、認証、タイマー、プライオリティなどの HSRP 属性を設定してあることを確認します。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*

3. **ipv6 address** *ipv6-address/length*
4. **hsrp version 2**
5. **hsrp** *group-number* **ipv6**
6. **ip** *ipv6-address*
7. **ip autoconfig**
8. **exit**
9. **no shutdown**
10. （任意） **show hsrp** [**group** *group-number*] [**ipv6**]
11. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 3/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**ipv6 address** *ipv6-address/length* 例： `switch(config-if)# ipv6 address 2001:0DB8::0001:0001/64`	インターフェイスの IPv6 アドレスを設定します。
Step 4	**hsrp version 2** 例： `switch(config-if-hsrp)# hsrp version 2`	HSRP バージョン 2 にこのグループを設定します。

Step 5	**hsrp** *group-number* **ipv6**
	例：
	`switch(config-if)# hsrp 10 ipv6`
	`switch(config-if-hsrp)#`
		IPv6 HSRP グループを作成し、HSRP コンフィギュレーション モードを開始します。HSRP バージョン 2 で指定できる範囲は 0 ～ 4095 です。デフォルト値は 0 です

Step 6	**ip** *ipv6-address* 例： `switch(config-if-hsrp)# ip 2001:DB8::1`	HSRP グループの仮想 IPv6 アドレスを設定し、そのグループをイネーブルにします。

	コマンドまたはアクション	目的
Step 7	**ip autoconfig** 例： `switch(config-if-hsrp)# ip autoconfig`	計算されたリンクローカル仮想 IPv6 アドレスから HSRP グループの仮想 IPv6 アドレスを自動設定し、グループをイネーブルにします。

Step 8	**exit**
	例：
	`switch(config-if-hsrp)# exit`
	`switch(config-if)#`
		HSRP設定モードを終了します。

Step 9	**no shutdown** 例： `switch(config-if)# no shutdown`	インターフェイスをイネーブルにします。
Step 10	（任意） **show hsrp** [**group** *group-number*] [**ipv6**] 例： `switch(config-if)# show hsrp group 10`	HSRP 情報を表示します。
Step 11	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	実行設定を、スタートアップ設定にコピーします。

例

（注） 設定完了後にインターフェイスを有効にするには、**no shutdown** コマンドを使用する必要があります。

次に Ethernet 3/2 上で IPv6 HSRP グループを設定する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 3/2
switch(config-if)# ipv6 address 2001:0DB8::0001:0001/64
switch(config-if-hsrp)# hsrp version 2
switch(config-if)# hsrp 2 ipv6
switch(config-if-hsrp)# ip 2001:DB8::1
switch(config-if-hsrp)# exit
switch(config-if)# no shutdown
switch(config-if)# copy running-config startup-config
```

HSRP 仮想 MAC アドレスの設定

設定されているグループ番号から HSRP が導き出したデフォルトの仮想 MAC アドレスを変更できます。

（注） vPC リンクの vPC ピアの両方で同じ仮想 MAC アドレスを設定する必要があります。

手順の概要

1. **mac-address** *string*
2. （任意） **hsrp use-bia [scope interface]**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**mac-address** *string* 例： `switch(config-if-hsrp)# mac-address 5000.1000.1060`	HSRP グループの仮想 MAC アドレスを設定します。ストリングには標準の MAC アドレス フォーマット（xxxx.xxxx.xxxx）を使用します。
Step 2	（任意） **hsrp use-bia [scope interface]** 例： `switch(config-if)# hsrp use-bia`	（注） 仮想 MAC アドレスに BIA（バーンドイン MAC アドレス）を使用するように HSRP を設定するには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。 HSRP 仮想 MAC アドレスにインターフェイスの BIA を使用するように、HSRP を設定します。**scope interface** キーワードを使用すると、このインターフェイス上のすべてのグループに BIA を使用するように HSRP を設定できます。

HSRP の認証

クリアテキストまたは MD5 ダイジェスト認証を使用してプロトコルを認証するように、HSRP を設定できます。MD5 認証はキーチェーンを使用します。詳細については、『Cisco Nexus 9000 Series NX-OS Security Configuration Guide』を参照してください。

始める前に

HSRP を有効にする必要があります（「HSRP の有効化」の項を参照）。

HSRP グループのすべてのメンバに同じ認証およびキーを設定したことを確認します。

MD5認証を使用している場合は、キーチェーンが作成されていることを確認します。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **hsrp** *group-number* [**ipv4** | **ipv6**]
4. **authentication** {**text** 文字列 | **md5** {**key-chain** キーチェーン | **key-string** {**0** | **7**} テキスト [**compatibility**] [**timeout** 秒]}}
5. （任意） **show hsrp** [**group** グループ数]
6. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**hsrp** *group-number* [**ipv4**	**ipv6**]
		例：
		`switch(config-if)# hsrp 2`
		`switch(config-if-hsrp)#`
			HSRP グループを作成し、HSRP設定モードを開始します。

Step 4	**authentication** {**text** 文字列	**md5** {**key-chain** キーチェーン	**key-string** {**0**	**7**} テキスト [**compatibility**] [**timeout** 秒]}}
				例：
				`switch(config-if-hsrp)# authentication text mypassword`
				例：
				`switch(config-if-hsrp)# authentication md5 key-chain hsrp-keys`
					authentication text コマンドを使用して、このインターフェイスに HSRP のクリアテキスト認証を設定します。または **authentication md5** コマンドを使用して、このインターフェイスに HSRP の MD5 認証を設定します。
					MD5 認証を設定する場合は、キーチェーンまたはキーストリングを使用できます。キーストリングを使用する場合は、必要に応じて、HSRP が新しいキーのみを受け入れる時間のタイムアウトを設定できます。範囲は 0 ～ 32767 秒です。
					互換性：Cisco IOS と Cisco NX-OS 間の認証の互換性のために設計されています。互換モードは MD5 キー文字列認証用です。非表示の認証タイプが Cisco IOS と Cisco NX-OS の両方で設定されている場合、HSRP

	コマンドまたはアクション	目的
		セッションを起動するには、NX-OS 側で互換性フラグを有効にする必要があります。
Step 5	（任意） **show hsrp** [**group** グループ数] 例： `switch(config-if-hsrp)# show hsrp group 2`	HSRP 情報を表示します。
Step 6	（任意） **copy running-config startup-config** 例： `switch(config-if-hsrp)# copy running-config startup-config`	実行設定を、スタートアップ設定にコピーします。

例

次に、キーチェーン作成後に HSRP の MD5 認証をイーサネット 1/2 上で設定する例を示します。

```
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 2013 23:59:59 Sep 12 2013
switch(config-keychain-key) send-lifetime 00:00:00 Jun 01 2013 23:59:59 Aug 12 2013
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 2013 23:59:59 Dec 12 2013
switch(config-keychain-key) send-lifetime 00:00:00 Sep 12 2013 23:59:59 Nov 12 2013
switch(config-keychain-key)# interface ethernet 1/2
switch(config-if)# hsrp 2
switch(config-if-hsrp)# authentication md5 key-chain hsrp-keys
switch(config-if-hsrp)# copy running-config startup-config
```

HSRP オブジェクト トラッキングの設定

他のインターフェイスまたはルータの可用性に基づいて、プライオリティが調整されるように HSRP グループを設定できます。スイッチがオブジェクト トラッキング対応として設定されていて、なおかつトラッキング対象のオブジェクトがダウンした場合、HSRP グループのプライオリティはダイナミックに変更されます。

トラッキング プロセスはトラッキング対象オブジェクトに定期的にポーリングを実行し、値の変化をすべて記録します。値が変化すると、HSRP がプライオリティを再計算します。HSRP インターフェイスにプリエンプションを設定している場合は、プライオリティの高い HSRP インターフェイスがアクティブ ルータになります。

手順の概要

1. **configure terminal**

2. **track** *object-id* **interface** *interface-type slot/port* {**line-protocol** | **ip routing** | **ipv6 routing**}
3. **track** *object-id* {**ip** | **ipv6**} **route** *ip-prefix/length* **reachability**
4. **exit**
5. **interface** *interface-type slot/port*
6. **hsrp** *group-number* [**ipv4** | **ipv6**]
7. **priority** [*value*]
8. **track** *object-id* [**decrement** *value*]
9. **preempt** [**delay** [**minimum** *seconds*] [**reload** *seconds*] [**sync** *seconds*]]
10. （任意） **show hsrp interface** *interface-type slot/port*
11. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**track** *object-id* **interface** *interface-type slot/port* {**line-protocol** | **ip routing** | **ipv6 routing**}
	例：
	`switch(config)# track 1 interface ethernet 2/2 line-protocol`
	`switch(config-track)#`
		トラック オブジェクトがトラッキングするインターフェイスを設定します。インターフェイスのステート変化は、次のようにトラック オブジェクトのステータスを左右します。
		• グローバルコンフィギュレーションモードで、**track** コマンドで使用するインターフェイスおよび対応するオブジェクト番号を設定します。
		• **line-protocol** キーワードを指定すると、インターフェイスがアップ状態かどうかが追跡されます。**ip routing** または **ipv6 routing** キーワードを指定すると、インターフェイス上でIPルーティングが有効であり、IPアドレスが設定されているかどうかもチェックされます。

Step 3	**track** *object-id* {**ip** | **ipv6**} **route** *ip-prefix/length* **reachability** 例： `switch(config-track)# track 2 ip route 192.0.2.0/8 reachability`	ルートのトラッキング対象オブジェクトを作成し、トラッキング コンフィギュレーション モードを開始します。*object-id* の範囲は 1 ～ 500 です。
Step 4	**exit** 例：	トラック コンフィギュレーション モードを終了します。

	コマンドまたはアクション	目的
	`switch(config-track)# exit` `switch(config)#`	

Step 5	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 6	**hsrp** *group-number* [**ipv4** | **ipv6**]
	例：
	`switch(config-if)# hsrp 2`
	`switch(config-if-hsrp)#`
		HSRP グループを作成し、HSRP設定モードを開始します。

Step 7	**priority** [*value*] 例： `switch(config-if-hsrp)# priority 254`	HSRP グループでのアクティブ ルータ選択に使用するプライオリティ レベルを設定します。有効な範囲は 0 ～ 255 です。デフォルトは 100 です。
Step 8	**track** *object-id* [**decrement** *value*] 例： `switch(config-if-hsrp)# track 1 decrement 20`	HSRP インターフェイスの重み付けを左右する、トラッキング対象のオブジェクトを指定します。 *value* 引数には、トラッキング対象のオブジェクトで障害が発生した場合に、HSRP インターフェイスのプライオリティから差し引く値を指定します。範囲は 1 ～ 255 です。デフォルトは 10 です。
Step 9	**preempt** [**delay** [**minimum** *seconds*] [**reload** *seconds*] [**sync** *seconds*]] 例： `switch(config-if-hsrp)# preempt delay minimum 60`	現在のアクティブ ルータよりプライオリティが高い場合に、HSRP グループのアクティブ ルータとして引き継ぐようにルータを設定します。このコマンドは、デフォルトでディセーブルになっています。任意で、遅延を設定して、HSRP グループのプリエンプションを設定した時間だけ遅らせることができます。指定できる範囲は 0 ～ 3600 秒です。
Step 10	（任意） **show hsrp interface** *interface-type slot/port* 例： `switch(config-if-hsrp)# show hsrp interface ethernet 1/2`	インターフェイスの HSRP 情報を表示します。
Step 11	（任意） **copy running-config startup-config** 例： `switch(config-if-hsrp)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、Ethernet インターフェイス 1/2 上で HSRP オブジェクト トラッキングを設定する例を示します。

```
switch# configure terminal
switch(config)# track 1 interface ethernet 2/2 line-protocol
switch(config-track)# track 2 ip route 192.0.2.0/8 reachability
switch(config-track)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# hsrp 2
switch(config-if-hsrp)# priority 254
switch(config-if-hsrp)# track 1 decrement 20
switch(config-if-hsrp)# preempt delay minimum 60
switch(config-if-hsrp)# copy running-config startup-config
```

HSRP プライオリティの設定

HSRP グループのプライオリティを設定できます。HSRP では、プライオリティを使用して、アクティブ ルータとして動作する HSRP グループ メンバを決定します。vPC 対応のインターフェイスで HSRP を設定する場合は、オプションで vPC トランクにフェールオーバーする時期を制御するしきい値の上限と下限を設定できます。スタンバイ ルータのプライオリティが下限のしきい値を下回った場合、HSRP は、すべてのスタンバイ ルータ トラフィックを vPC トランク全体に送信し、アクティブな HSRP ルータを通して転送します。HSRP では、スタンバイ HSRP ルータ プライオリティが上限しきい値を超えるまで、この状況を維持します。

IPv6 HSRP グループでは、すべてのグループ メンバのプライオリティが同じ場合、HSRP は IPv6 リンクローカル アドレスに基づいてアクティブ ルータを選択します。

HSRP プライオリティを設定するには、インターHSRP グループ設定モードで次のコマンドを使用します。

手順の概要

1. **priority** *level* [**forwarding-threshold lower** *lower-value* **upper** *upper-value*]

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**priority** *level* [**forwarding-threshold lower** *lower-value* **upper** *upper-value*] 例： `switch(config-if-hsrp)# priority 60 forwarding-threshold lower 40 upper 50`	HSRP グループでのアクティブ ルータ選択に使用するプライオリティ レベルを設定します。*level* の範囲は 0 ～ 255 です。デフォルトは 100 です。オプションで、このコマンドを使用して vPC トランクにフェールオーバーする時点を決定するために vPC が使用するしきい値の上限と下限を設定できます。*lower-value* の範囲は 1 ～ 255 です。デフォルトは 1 です。

	コマンドまたはアクション	目的
		upper-value の範囲は 1 ～ 255 です。デフォルトは 255 です。

HSRP コンフィギュレーション モードでの HSRP のカスタマイズ

必要に応じて、HSRP の動作をカスタマイズできます。仮想 IP アドレスを設定することによって、HSRP グループをイネーブルにすると、そのグループがただちに動作可能になることに注意してください。HSRP をカスタマイズする前に HSRP グループをイネーブルにした場合、機能のカスタマイズが完了しないうちに、ルータがグループの制御を引き継いでアクティブ ルータになる可能性があります。HSRP のカスタマイズを予定している場合は、HSRP グループをイネーブルにする前に行ってください。

手順の概要

1. （任意） **name** *string*
2. （任意） **preempt** [**delay** [**minimum** *seconds*] [**reload** *seconds*] [**sync** *seconds*]]
3. （任意） **timers** [**msec**] *hellotime* [**msec**] *holdtime*
4. （任意） **hsrp delay minimum** *seconds*
5. （任意） **hsrp delay reload** *seconds*

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	（任意） **name** *string* 例： `switch(config-if-hsrp)# name HSRP-1`	HSRP グループの IP 冗長名を指定します。*string* は 1 ～ 255 文字です。デフォルトの文字列の形式は、hsrp-*interface short-name group-id*です。たとえば、hsrp-Eth2/1-1 です。
Step 2	（任意） **preempt** [**delay** [**minimum** *seconds*] [**reload** *seconds*] [**sync** *seconds*]] 例： `switch(config-if-hsrp)# preempt delay minimum 60`	現在のアクティブ ルータよりもプライオリティが高い場合に、HSRP グループのアクティブ ルータとして引き継ぐようにルータを設定します。このコマンドは、デフォルトでディセーブルになっています。任意で、遅延を設定して、HSRP グループのプリエンプションを設定した時間だけ遅らせることができます。指定できる範囲は 0 ～ 3600 秒です。
Step 3	（任意） **timers** [**msec**] *hellotime* [**msec**] *holdtime* 例：	次のように、この HSRP メンバーの hello タイムおよびホールド タイムを設定します。

	コマンドまたはアクション	目的
	`switch(config-if-hsrp)# timers 5 18`	• *hellotime*：hello パケットを送信してから、次の hello パケットを送信するまでのインターバル。指定できる範囲は 1 ～ 254 秒です。 • *holdtime*：hello パケットの情報が無効と見なされるまでのインターバル。指定できる範囲は 3 ～ 255 です。 オプションの **msec** キーワードは、引数がデフォルトの秒単位ではなく、ミリ秒単位で表されることを指定します。タイマーの範囲（ミリ秒）は次のとおりです。 • *hellotime*：hello パケットを送信してから、次の hello パケットを送信するまでのインターバル。指定できる範囲は 250 ～ 999 ミリ秒です。 • *holdtime*：hello パケットの情報が無効と見なされるまでのインターバル。指定できる範囲は 750 ～ 3000 ミリ秒です。
Step 4	（任意） **hsrp delay minimum** *seconds* 例： `switch(config-if)# hsrp delay minimum 30`	グループがイネーブルになってから、グループに参加するまでに HSRP が待機する最小時間を指定します。指定できる範囲は 0 ～ 10000 秒です。デフォルトは 0 です。
Step 5	（任意） **hsrp delay reload** *seconds* 例： `switch(config-if)# hsrp delay reload 30`	リロード後、グループに参加するまでに HSRP が待機する最小時間を指定します。指定できる範囲は 0 ～ 10000 秒です。デフォルトは 0 です。 （注） [リロード（reload）] オプションでプリエンプト遅延を使用する場合は、**hsrp delay reload**（インターフェイスレベルコマンド）とともに使用することをお勧めします。これは、リロード後 SVI がアップしていても物理リンク/ポートチャネルがリロード後にまだアップしていないため、プリエンプション遅延リロードタイマーが開始しなかったために、優先順位の高い HSRP スタンバイがホールドタイマーの期限切れ（10秒）でアクティブになるシナリオを回避するためです。タイマーは規模に応じて調整できます。 例：**preempt delay reload 200** 構成する代わりに、**preempt delay reload 140** および **hsrp delay reload 60** を構成します。これは、リロード遅延の有効期限

	コマンドまたはアクション	目的
		（60 秒）後に HSRP がマシンの起動を INIT 状態から開始するときに、SVI と物理リンク/ポートチャネルの両方がアップ状態になるようにするためです。

インターフェイスコンフィギュレーションモードでのHSRPのカスタマイズ

必要に応じて、HSRPの動作をカスタマイズできます。仮想IPアドレスを設定することによって、HSRP グループをイネーブルにすると、そのグループがただちに動作可能になることに注意してください。HSRP をカスタマイズする前に HSRP グループをイネーブルにした場合、機能のカスタマイズが完了しないうちに、ルータがグループの制御を引き継いでアクティブ ルータになる可能性があります。HSRPのカスタマイズを予定している場合は、HSRPグループをイネーブルにする前に行ってください。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **hsrp delay minimum** *seconds*
4. **hsrp delay reload** *seconds*
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**hsrp delay minimum** *seconds* 例： `switch(config-if)# hsrp delay minimum 30`	グループがイネーブルになってから、グループに参加するまでに HSRP が待機する最小時間を指定します。指定できる範囲は 0 ～ 10000 秒です。デフォルトは 0 です。

	コマンドまたはアクション	目的
Step 4	**hsrp delay reload** *seconds* 例： `switch(config-if)# hsrp delay reload 30`	リロード後、グループに参加するまでに HSRP が待機する最小時間を指定します。指定できる範囲は 0 ～ 10000 秒です。デフォルトは 0 です。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

HSRP の拡張ホールド タイマーの設定

制御された（グレースフル）スイッチオーバー中に拡張 NSF をサポートするために拡張ホールド タイマーを使用するように HSRP を設定できます。拡張ホールド タイマーは、すべての HSRP ルータ上で設定してください

（注） 拡張ホールド タイマーを設定する場合は、すべての HSRP ルータで拡張ホールド タイマーを設定する必要があります。デフォルトでないホールド タイマーを設定する場合は、HSRP 拡張ホールド タイマーの設定時にすべての HSRP ルータで同じ値を設定してください。

（注） HSRP 拡張ホールド タイマーは、HSRPv1 のミリ秒の hello タイマーやホールド タイマーを設定した場合は適用されません。これは、HSRPv2 には適用されません。

手順の概要

1. （任意） **hsrp timers extended-hold** [*timer*]
2. （任意） **show hsrp**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	（任意） **hsrp timers extended-hold** [*timer*] 例： `switch(config)# hsrp timers extended-hold`	IPv4 と IPv6 の両方のグループに、HSRP 拡張ホールド タイマーを秒単位で設定します。タイマーの範囲は 10 ～ 255 です。デフォルトは 10 です。 （注）

	コマンドまたはアクション	目的
		拡張ホールド時間を表示するには、**show hsrp** コマンドまたは **show running-config hsrp** コマンドを使用します。
Step 2	（任意） **show hsrp** 例： `switch(config)# show hsrp`	HSRP 拡張ホールド タイムを表示します。

例

拡張ホールド タイムを表示するには、**show hsrp** コマンドまたは **show running-config hsrp** コマンドを使用します。

HSRP 設定の確認

HSRP 設定情報を表示するには、次のいずれかの作業を実行します。

コマンド	目的
show hsrp [**group** *group-number*]	すべてのグループまたは特定のグループのHSRPステータスを表示します。
show hsrp delay [**interface** *interface-type slot/port*]	すべてのインターフェイスまたは特定のインターフェイスの HSRP 遅延値を表示します。
show hsrp [**interface** *interface-type slot/port*]	インターフェイスの HSRP ステータスを表示します。
show hsrp [**group** *group-number*] [**interface** *interface-type slot/port*] [**active**] [**all**] [**init**] [**learn**] [**listen**] [**speak**] [**standby**]	ステートが active、init、listen、または standby の仮想フォワーダについて、グループまたはインターフェイスの HSRP ステータスを表示します。disabled を含めてすべてのステートを表示する場合は、**all** キーワードを使用します。
show hsrp [**group** *group-number*] [**interface** *interface-type slot/port*] [**active**] [**all**] [**init**] [**learn**] [**listen**] [**speak**] [**standby**] **brief**	ステートが active、init、listen、または standby の仮想フォワーダについて、グループまたはインターフェイスの HSRP ステータスの要約を表示します。disabled を含めてすべてのステートを表示する場合は、**all** キーワードを使用します。

コマンド	目的
show ip local-pt	ネットスタックが VIP サブネットのサブネットルートをプログラムしているかどうかを表示します。

HSRP の設定例

次に、MD5 認証およびインターフェイス トラッキングを指定して、インターフェイス上で HSRP をイネーブルにする例を示します。

```
key chain hsrp-keys
key 0
key-string 7 zqdest
accept-lifetime 00:00:00 Jun 01 2013 23:59:59 Sep 12 2013
send-lifetime 00:00:00 Jun 01 2013 23:59:59 Aug 12 2013
key 1
key-string 7 uaeqdyito
accept-lifetime 00:00:00 Aug 12 2013 23:59:59 Nov 12 2013
send-lifetime 00:00:00 Sep 12 2013 23:59:59 Nov 12 2013

feature hsrp
track 2 interface ethernet 2/2 ip
interface ethernet 1/2
ip address 192.0.2.2/8
hsrp 1
authenticate md5 key-chain hsrp-keys
priority 90
track 2 decrement 20
ip 192.0.2.10
no shutdown
```

次の例は、インターフェイスに HSRP プライオリティを設定する方法を示しています。

```
interface vlan 1
hsrp 0
preempt
priority 100 forwarding-threshold lower 80 upper 90
ip 192.0.2.2
track 1 decrement 30
```

次に、インターフェイス IP アドレスのサブネットとは異なるサブネットに設定された HSRP サブネット VIP アドレスを設定する例を示します。

```
sswitch# configure terminal
switch(config)# feature hsrp
switch(config)# feature interface-vlan
switch(config)# interface vlan 2
switch(config-if)# ip address 192.0.2.1/24
switch(config-if)# hsrp 2
switch(config-if-hsrp)# ip 209.165.201.1/24
```

次に、インターフェイス IP アドレスのサブネットとは異なるサブネットに設定された HSRP サブネット VIP アドレスを設定する例を示します。

```
switch# configure terminal
switch(config)# feature hsrp
switch(config)# feature interface-vlan
switch(config)# interface vlan 2
switch(config-if)# ip address 192.0.2.1/24
switch(config-if)# hsrp 2
switch(config-if-hsrp)# ip 209.165.201.1
!ERROR: VIP subnet mismatch with interface IP!
```

次の例は、HSRP サブネットの VIP アドレスがインターフェイス IP アドレスと同じサブネットに設定されている場合の VIP の不一致エラーを示しています。

```
switch# configure terminal
switch(config)# feature hsrp
switch(config)# feature interface-vlan
switch(config)# interface vlan 2
switch(config-if)# ip address 192.0.2.1/24
switch(config-if)# hsrp 2
switch(config-if-hsrp)# ip 192.0.2.10/24
!ERROR: Subnet VIP cannot be in same subnet as interface IP!
```

その他の参考資料

HSRP の実装に関する詳細は、次の各項を参照してください。

- 関連資料
- MIB

関連資料

関連項目	マニュアル タイトル
VRRP の設定	『Configuring VRRP』
高可用性の設定	『Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide』

MIB

MIB	MIB のリンク
HSRP に関連する MIB	サポートされている MIB を検索およびダウンロードするには、次の URL にアクセスしてください。 ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

第 20 章

VRRP の設定

この章は、次の項で構成されています。

- VRRP について （681 ページ）
- VRRPv3およびVRRSに関する情報 （687 ページ）
- 高可用性 （688 ページ）
- 仮想化のサポート （688 ページ）
- VRRP の注意事項と制約事項 （689 ページ）
- VRRPv3 の注意事項および制約事項 （689 ページ）
- VRRP パラメータのデフォルト設定 （690 ページ）
- VRRPv3 パラメータのデフォルト設定 （691 ページ）
- VRRP の設定 （691 ページ）
- VRRPv3 の設定 （703 ページ）
- VRRP の設定の確認 （711 ページ）
- VRRPv3 設定の確認 （711 ページ）
- VRRP 統計情報のモニタリングとクリア （712 ページ）
- VRRPv3 統計情報のモニタリングとクリア （712 ページ）
- VRRP の設定例 （712 ページ）
- VRRPv3 の設定例 （714 ページ）
- その他の参考資料 （715 ページ）

VRRP について

VRRPを使用すると、仮想IPアドレスを共有するルータ グループを設定することによって、ファーストホップ IP ルータで透過的フェールオーバーが可能になります。VRRP ではそのグループに許可されるルータが選択され、仮想 IP アドレスへのすべてのパケットが処理できるようになります。残りのルータはスタンバイになり、許可されるルータで障害が発生した場合に処理を引き継ぎます。

VRRP の動作

LAN クライアントは、ダイナミック プロセスまたはスタティック設定を使用することによって、特定のリモート宛先へのファーストホップにするルータを決定できます。ダイナミック ルータ ディスカバリの例を示します。

プロキシ ARP：クライアントはアドレス解決プロトコル（ARP）を使用して到達すべき宛先を取得します。ルータは独自の MAC アドレスで ARP 要求に応答します。

ルーティング プロトコル：クライアントはダイナミック ルーティング プロトコルのアップデートを（ルーティング情報プロトコル（RIP）などから）受信し、独自のルーティング テーブルを形成します。

ICMP Router Discovery Protocol（IRDP）クライアント：クライアントはインターネット制御メッセージ プロトコル（ICMP）ルータ ディスカバリ クライアントを実行します。

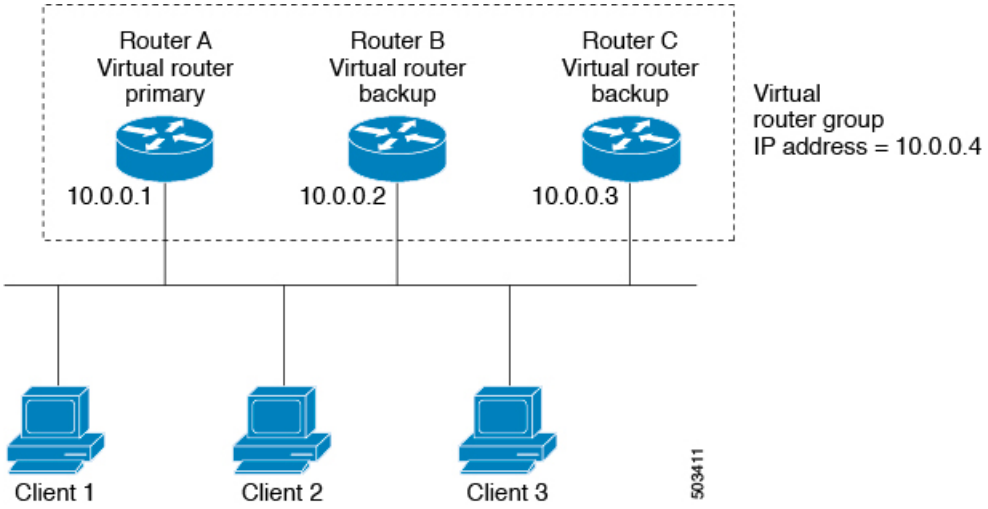
ダイナミック ディスカバリ プロトコルのデメリットは、LAN クライアントにある程度、設定および処理のオーバーヘッドが発生することです。また、ルータが故障した場合、他のルータに切り替えるプロセスも遅くなる場合があります。

ダイナミック ディスカバリ プロトコルの代わりに、クライアント上でデフォルト ルータをスタティックに設定することもできます。このアプローチでは、クライアントの設定および処理が簡素化されますが、シングルポイント障害が生じます。デフォルト ゲートウェイで障害が発生した場合、LAN クライアントの通信はローカル IP ネットワーク セグメントに限定され、ネットワークの他の部分から切り離されます。

VRRP では、ルータ グループ（VRRP グループ）が単一の仮想 IP アドレスを共有できるようにすることによって、スタティック設定に伴う問題を解決できます。さらに、デフォルト ゲートウェイとして仮想 IP アドレスを指定して、LAN クライアントを設定できます。

次の図は、基本的な VLAN トポロジです。この例では、ルータ A、B、および C が VRRP グループを形成します。グループの IP アドレスは、ルータ A のインターフェイス インターフェイスに設定されているアドレス（10.0.0.1）と同じです。

図 *46*：基本的な *VRRP* トポロジ



仮想 IP アドレスにルータ A の物理イーサネット インターフェイスの IP アドレスが使用されるので、ルータ A がプライマリ（「IP アドレス オーナー」）になります。ルータ A はプライマリとして、VRRP グループの仮想 IP アドレスを所有し、送信されたパケットをこの IP アドレスに転送します。クライアント 1 ～ 3 には、デフォルト ゲートウェイの IP アドレス 10.0.0.1 が設定されています。

ルータ B および C の役割はバックアップです。プライマリで障害が発生すると、プライオリティが最も高いバックアップ ルータがプライマリになり、仮想 IP アドレスを引き継いで、LAN ホストへのサービスが途切れないようにします。ルータ A が回復すると、これが再びプライマリになります。

（注） ルーテッド ポートで受信した VRRP 仮想 IP アドレス宛のパケットは、ローカル ルータ上で終端します。そのルータがプライマリ VRRP ルータであるのかバックアップ VRRP ルータであるのかは関係ありません。これらのパケットには、ping トラフィックと Telnet トラフィックが含まれます。レイヤ 2（VLAN）インターフェイスで受信した、VRRP 仮想 IP アドレス宛のパケットは、プライマリ ルータに届きます。

VRRP の利点

VRRP の利点は、次のとおりです。

- 冗長性：複数のルータをデフォルト ゲートウェイ ルータとして設定できるので、ネットワークにシングル ポイント障害が発生する確率が下がります。
- ロード シェアリング：複数のルータで LAN クライアントとの間のトラフィックを分担できます。トラフィックの負荷が使用可能なルータ間でより公平に分担されます。
- マルチ VRRP グループ：プラットフォームが複数の MAC アドレスをサポートする場合、ルータの物理インターフェイス上で、複数の VRRP グループをサポートします。マルチ VRRP グループによって、LAN トポロジで冗長性およびロード シェアリングを実現できます。
- マルチ IP アドレス：セカンダリ IP アドレスを含めて、複数の IP アドレスを管理できます。イーサネット インターフェイス上で複数のサブネットを設定している場合は、各サブネットで VRRP を設定できます。
- プリエンプト：障害プライマリを引き継いでいたバックアップ ルータより、さらにプライオリティが高いバックアップ ルータが使用可能になったときに、プライオリティが高い方を優先させることができます。
- アドバタイズメント プロトコル：VRRP アドバタイズメントに、専用のインターネット割り当て番号局（IANA）規格マルチキャスト アドレス（224.0.0.18）を使用します。このアドレッシング方式によって、マルチキャストを提供するルータ数が最小限になり、テスト機器でセグメント上の VRRP パケットを正確に識別できるようになります。IANA は VRRP に IP プロトコル番号 112 を割り当てています。
- VRRP トラッキング：インターフェイスのステートに基づいて VRRP プライオリティを変更することによって、最適な VRRP ルータがグループのプライマリになることが保証されます。

複数の VRRP グループ

物理インターフェイス上で複数の VRRP グループを設定できます。サポートされる VRRP グループの数については、『Cisco Nexus 9000 Series NX-OS Verified Scalability Guide』を参照してください。

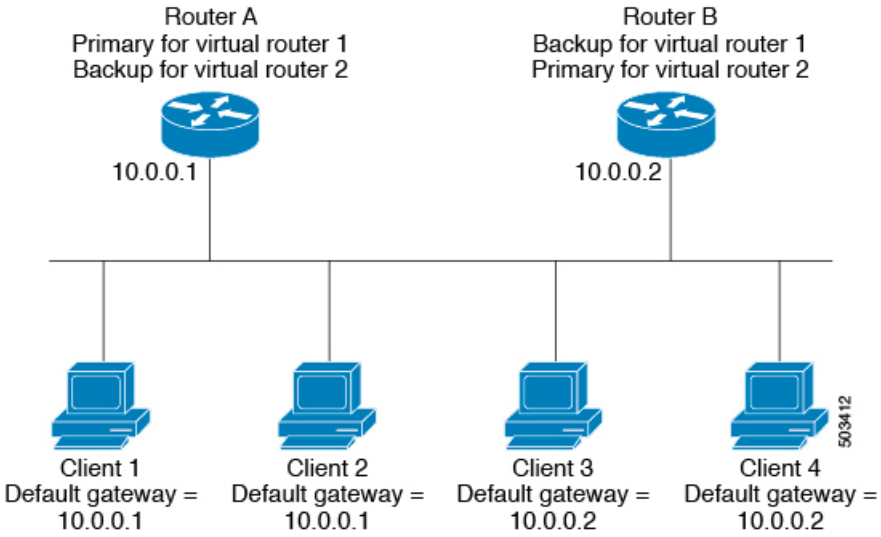
ルータ インターフェイスがサポートできる VRRP グループの数は、次の要因によって決まります。

- ルータの処理能力
- ルータのメモリの能力

ルータ インターフェイス上で複数の VRRP グループが設定されたトポロジでは、インターフェイスはある VRRP グループのプライマリ、および他の 1 つまたは複数の VRRP グループのバックアップとして動作可能です。

次の図の LAN トポロジでは、ルータ A と B がクライアント 1 ～ 4 のトラフィックを共有するように、VRRP が設定されています。ルータ A と B の一方で障害が発生した場合、もう一方がバックアップとして機能します。

図 *47*: ロード シェアリングおよび冗長構成の *VRRP* トポロジ



このトポロジには、オーバーラップする 2 つの VRRP グループに対応する 2 つの仮想 IP アドレスが含まれています。VRRP グループ 1 では、ルータ A が IP アドレス 10.0.0.1 のオーナーであり、プライマリです。ルータ B はルータ A をバックアップします。クライアント 1 と 2 には、デフォルト ゲートウェイの IP アドレス 10.0.0.1 が設定されています。

VRRP グループ 2 では、ルータ B が IP アドレス 10.0.0.2 のオーナーであり、プライマリです。ルータ A はルータ B をバックアップします。クライアント 3 と 4 には、デフォルト ゲートウェイの IP アドレス 10.0.0.2 が設定されています。

VRRP ルータのプライオリティおよびプリエンプション

VRRP 冗長構成の重要な側面は、VRRP ルータのプライオリティです。各 VRRP ルータが果たす役割やプライマリ ルータで障害が発生した場合のアクションは、プライオリティによって決まるからです。

VRRP ルータが仮想 IP アドレスおよび物理インターフェイスの IP アドレスを所有する場合、そのルータはプライマリとして機能します。プライマリのプライオリティは 255 です。

プライオリティによって、VRRP ルータがバックアップ ルータとして動作するかどうかが決まり、さらに、プライマリで障害が発生した場合にプライマリになる順序も決まります。

たとえば、ルータ A が LAN トポロジにおけるプライマリであり、そのルータ A で障害が発生した場合、VRRP はバックアップ B が引き継ぐのか、バックアップ C が引き継ぐのかを判断する必要があります。ルータ B にプライオリティ 101 が設定されていて、ルータ C がデフォルトのプライオリティ 100 の場合、VRRP はルータ B をプライマリになるべきルータとして選択します。ルータ B の方がプライオリティが高いからです。ルータ B および C にデフォルトのプライオリティ 100 が設定されている場合は、VRRP は IP アドレスが大きい方のバックアップをプライマリになるべきルータとして選択します。

VRRP ではプリエンプションを使用して、VRRP バックアップ ルータがプライマリになってからのアクションを決定します。プリエンプションはデフォルトでイネーブルなので、VRRP は新しいプライマリよりプライオリティの高いバックアップがオンラインになると、バックアップに切り替えます。たとえば、ルータ A がプライマリであり、そのルータ A で障害が発生した場合、VRRP は（プライオリティの順位が次である）ルータ B を選択します。ルータ C がルータ B より高いプライオリティでオンラインになると、ルータ B で障害が発生していなくても、VRRP はルータ C を新しいプライマリとして選択します。

プリエンプションを無効にした場合、VRRP が切り替わるのは、元のプライマリが回復した場合、または新しいプライマリで障害が発生した場合に限られます。

vPC と VRRP

VRRP は仮想ポート チャネル（vPC）と相互運用できます。vPC を使用すると、2 個の異なる Cisco Nexus 9000 シリーズ スイッチを物理的に接続し、第 3 のデバイスからは 1 つのポートとして見えるリンクが実現します。vPC の詳細については、『Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide』を参照してください。

vPC はプライマリ VRRP ルータとバックアップ VRRP ルータの両方を使用してトラフィックを転送します。「VRRP プライオリティの設定」のセクションを参照してください。

（注） プライマリ vPC ピア デバイスの VRRP をアクティブに、セカンダリ vPC デバイスの VRRP をスタンバイにそれぞれ設定する必要があります。

VRRP のアドバタイズメント

VRRP プライマリは、同じグループ内の他の VRRP ルータに VRRP アドバタイズメントを送信します。アドバタイズメントは、プライマリのプライオリティと状態を伝えます。Cisco NX-OS は、VRRPアドバタイズメントを IP パケットにカプセル化し、VRRPグループに割り当てられた IP マルチキャストアドレスに送信します。デフォルトでは、Cisco NX-OS が 1 秒ごとにアドバタイズメントを送信しますが、異なるアドバタイズメント間隔を設定できます。

VRRP 認証

VRRP は、次の認証機能をサポートします。

- 認証なし
- プレーン テキスト認証

VRRP は次の場合に、パケットを拒否します。

- 認証方式がルータと着信パケットで異なる。
- テキスト認証文字列がルータと着信パケットで異なる。

VRRP トラッキング

VRRP は次のトラッキング オプションをサポートしています。

- ネイティブインターフェイストラッキング：インターフェイスのステートを追跡し、そのステートを使用して VRRP グループの VRRP ルータのプライオリティを判別します。インターフェイスがダウンしている場合、またはインターフェイスにプライマリ IP アドレスがない場合、トラッキング対象ステートはダウンとなります。
- オブジェクト トラッキング：設定されたオブジェクトのステートを追跡し、そのステートを使用して VRRP グループの VRRP ルータのプライオリティを判別します。オブジェクト トラッキングの詳細については、「オブジェクト トラッキングの設定」を参照してください。

トラッキング対象ステート（インターフェイスまたはオブジェクト）がダウンになると、VRRP はユーザがトラッキング対象ステートに対して新しいプライオリティをどのように設定するかに基づいて、プライオリティをアップデートします。トラッキング対象ステートがオンラインになると、VRRP は仮想ルータ グループの元のプライオリティを復元します。

たとえば、ネットワークへのアップリンクがダウンした場合、別のグループ メンバーが VRRP グループのプライマリとして引き継げるように、VRRP グループ メンバーのプライオリティを引き下げなければならないことがあります。詳細については、「VRRP インターフェイス ステート トラッキングの設定」の項を参照してください。

（注） VRRP はレイヤ 2 インターフェイスのトラッキングをサポートしていません。

VRRP 用 BFD

この機能では、双方向フォワーディング検出（BFD）をサポートします。BFD は、高速転送とパス障害の検出時間を提供する検出プロトコルです。BFD は 2 台の隣接デバイス間のサブセカンド障害を検出し、BFD の負荷の一部を、サポートされるモジュール上のデータ プレーンに分散できるため、プロトコル hello メッセージよりも CPU を使いません。詳細については、『Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide』を参照してください。

VRRPv3およびVRRSに関する情報

VRRP のバージョン 3（VRRPv3）では、スイッチのグループで単一の仮想スイッチを形成して、冗長性を実現し、ネットワーク内のシングル ポイント障害が生じる可能性を減らすことができます。これにより、仮想スイッチをデフォルト ゲートウェイとして使用するように、LAN クライアントを設定できます。スイッチのグループを表す仮想スイッチは、VRRPv3 グループとも呼ばれます。

仮想ルータ冗長サービス（VRRS）では、VRRPv3 を監視することでステートレス冗長サービスを VRRS 経路と VRRS クライアントに提供することで VRRPv3 のスケーラビリティが向上します。VRRPv3 は、VRRPv3 ステータス情報（現在および過去の冗長状態、アクティブおよび非アクティブのレイヤ 2 およびレイヤ 3 アドレスなど）を VRRS 経路とすべての登録済み VRRS クライアントに配信する VRRS サーバとして機能します。

VRRS クライアントは、VRRPv3 を使用して、グループのステートに応じてサービスやリソースを提供または抑制する他の Cisco プロセスまたはアプリケーションです。VRRS 経路は、VRRS データベース情報を使用して、拡張インターフェイス環境全体に拡張ファーストホップ ゲートウェイの冗長性を提供する特殊な VRRS クライアントです。

VRRS は、自身の状態を維持することが制限されています。VRRPv3 グループに VRRS クライアントをリンクすると、ステートレスまたはステートフル フェールオーバーが実装可能になるように、VRRS でクライアント アプリケーションにサービスを提供できるようにするメカニズムが提供されます。ステートフル フェールオーバーでは、フェールオーバーが発生したときに運用データが失われないように障害の前に所定バックアップとの通信が必要になります。

VRRS 経路はクライアントと同様に動作しますが、VRRS アーキテクチャと統合されます。この経路により、何百ものインターフェイス間で1つの仮想アドレスを設定することでファーストホップ ゲートウェイの冗長性を拡張する方法が提供されます。VRRS 経路の仮想ゲートウェイの状態は、ファーストホップ冗長プロトコル（FHRP）VRRS サーバの状態によります。

VRRPv3 は、現在の状態（プライマリ、バックアップ、または運用不可能な初期状態（INIT））を VRRS に通知し、その情報を経路またはクライアントに渡します。VRRPv3 グループ名は、VRRS をアクティブにし、VRRPv3 グループをクライアントまたは同じ名前の VRRS の一部として設定されている経路と関連付けます。

経路およびクライアントは、VRRPv3 サーバの状態で機能します。VRRPv3 グループの状態が変化すると、VRRS経路とクライアントの動作（インターフェイスのシャットダウン、アカウンティング ログの追加などのタスクの実行）が VRRS から受信した状態により変化します。

VRRPv3 の利点

VRRPv3の利点は次のとおりです。

- マルチベンダー環境での相互運用性
- IPv4およびIPv6アドレスファミリのサポート
- VRRS 経路によるスケーラビリティの向上

VRRPv3 オブジェクト トラッキング

Cisco NX-OS リリース 9.2(2) 以降、VRRPv3 はオブジェクト トラッキングをサポートしています。この機能は、設定されたオブジェクトの状態を追跡し、その状態を使用して VRRPv3 グループの VRRPv3 ルータの優先順位を判別します。オブジェクト トラッキングの詳細については、「オブジェクト トラッキングの設定」を参照してください。

トラッキング対象オブジェクトがダウンすると、VRRPv3 は設定された値だけ優先順位を引き下げます。デフォルト値は10です。同じトラッキング対象オブジェクトが再びダウンした場合、アクションは実行されません。トラッキング対象オブジェクトがアップになると、VRRPv3 は設定された値だけ優先順位を上げます。

（注） VRRPv3 は、レイヤ 2 インターフェイスのトラッキングまたはネイティブ インターフェイスのトラッキングをサポートしていません。

高可用性

VRRP は、ステートフル リスタートとステートフル スイッチオーバーを通して高可用性をサポートします。ステートフル リスタートは、VRRP が障害を処理してリスタートするときに行われます。ステートフル スイッチオーバーは、アクティブ スーパーバイザがスタンバイ スーパーバイザに切り替わるときに行われます。Cisco NX-OS は、スイッチオーバー後に実行コンフィギュレーションを適用します。

VRRPv3 は、ステートフル スイッチオーバーをサポートしていません。

仮想化のサポート

VRRP は、仮想ルーティングおよび転送（VRF）インスタンスをサポートします。

VRRP の注意事項と制約事項

VRRP には、次の注意事項および制限事項があります。

- 管理インターフェイス上で VRRP を設定できません。
- VRRP がイネーブルの場合は、ネットワーク上のデバイス全体で VRRP 設定を複製する必要があります。
- 同一インターフェイス上では、複数のファーストホップ冗長プロトコルを設定しないことを推奨します。
- VRRP を設定するインターフェイスに IP アドレスを設定し、そのインターフェイスをイネーブルにしてからでなければ、VRRP はアクティブになりません。
- インターフェイス VRF メンバーシップまたはポート チャネル メンバーシップを変更した場合、またはポート モードをレイヤ 2 に変更した場合は、Cisco NX-OS によってインターフェイス上のすべてのレイヤ 3 設定が削除されます。
- VRRP でレイヤ 2 インターフェイスを追跡するよう設定した場合、レイヤ 2 をシャットダウンしてからインターフェイスを再度イネーブル化することにより、VRRP プライオリティを更新してレイヤ 2 インターフェイスのステートを反映させる必要があります。

  VRRP の BFD は、2 台のルータ間でのみ設定できます。

VRRPv3 の注意事項および制約事項

VRRPv3 設定時の注意事項および制約事項は、次のとおりです。

- リリース 9.3(1) では、VRRPv3 機能は、-R ライン カードを備えた Cisco Nexus 9504、9508、および 9516 スイッチで、最大 4095 の VRRPv3 グループと VRRS 経路をサポートします。
- VRRPv3 は既存のダイナミック プロトコルの代替にはなりません。VRRPv3 は、マルチアクセス、マルチキャスト、または ブロードキャスト対応イーサネット LAN で使用するために設計されています。
- VRRPv3は、イーサネットおよびファストイーサネットインターフェイス、ブリッジグループ仮想インターフェイス（BVI）、ギガビットイーサネットインターフェイス、およびVLANでのみサポートされます。
- VRRPv3 が使用中の場合、VRRPv2 は使用できません。VRRPv3 を設定するには、VRRPv2 設定を無効にする必要があります。
- VRRS は現在、VRRPv3 と合わせて使用する場合にのみ使用できます。
- VRRPv3 ミリ秒タイマーは、絶対に必要な場合以外は使用しないようにし、使用する場合は慎重な検討とテストが必要です。ミリ秒の値は望ましい状況でのみ動作します。ミリ秒のタ

イマー値は、VRRPv3 も含めてサポートしている限り、サードパーティ ベンダーと互換性があります。

- VRRPv3 が VRRS 経路の冗長インターフェイスと同じネットワーク パス上で動作する場合にのみ、完全なネットワークの冗長性を実現できます。完全な冗長性のために、次の制約事項が適用されます。
  - VRRS 経路は、親 VRRPv3 グループと同じ物理インターフェイスを使用する必要があるか、または親 VRRPv3 グループと同じ物理インターフェイスを持つサブインターフェイス上で設定する必要があります。
  - VRRS 経路をスイッチ仮想インターフェイス（SVI）に設定できるのは、関連付けられた VLAN が親 VRRPv3 グループが設定された VLAN と同じトランクを共有する場合のみです。
- VRRPv2とは異なり、VRRPv3は障害検出を高速化するための双方向転送をサポートしていません。
- VRRPv2 とは異なり、VRRPv3 はネイティブ インターフェイス トラッキングをサポートしていません。
- オブジェクト トラッキングを設定する前に、オブジェクトを作成する必要があります。
- VRRPv3 オブジェクト トラッキングには、次の注意事項と制限事項が適用されます。
  - Cisco NX-OS リリース9.2(2) 以降、すべての Cisco Nexus 9000 シリーズ スイッチおよびラインカードで、VRRPv3 オブジェクト トラッキングがサポートされます。
  - vPC ドメインでは VRRPv3 オブジェクト トラッキングを使用しないことを推奨します。

VRRP パラメータのデフォルト設定

次の表に、VRRP パラメータのデフォルト設定を示します。

表 *31 :* デフォルトの *VRRP* パラメータ

パラメータ	デフォルト
VRRP	ディセーブル
アドバタイズ インターバル	1 秒
認証	認証なし
プリエンプション	イネーブル
プライオリティ	100

VRRPv3 パラメータのデフォルト設定

次の表に、VRRPv3 パラメータのデフォルト設定を示します。

表 *32 : VRRPv3* のデフォルト パラメータ

パラメータ	デフォルト
VRRPv3	ディセーブル
VRRS	ディセーブル
VRRPv3 セカンダリ アドレスの一致	イネーブル
VRRPv3 グループのプライオリティ	100
VRRPv3 アドバタイズメント タイマー	1000 ミリ秒

VRRP の設定

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

VRRP のイネーブル化

VRRP グループを設定してイネーブルにするには、事前に VRRP 機能をグローバルにイネーブルにしておく必要があります。

手順の概要

1. **configure terminal**
2. [**no**] **feature vrrp**
3. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例:	グローバル設定モードを開始します。

	コマンドまたはアクション	目的
	`switch# configure terminal` `switch(config)#`	
Step 2	[**no**] **feature vrrp** 例： `switch(config)# feature vrrp`	VRRP をイネーブルにします。VRRP をディセーブルにするには、このコマンドの **no** 形式を使用します。
Step 3	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

VRRP グループの設定

VRRP グループを作成し、仮想 IP アドレスを割り当て、グループを有効にすることができます。

VRRP グループに設定できる仮想 IPv4 アドレスは 1 つです。プライマリ VRRP ルータはデフォルトで、仮想 IP アドレスを直接の宛先とするパケットをドロップします。これは、VRRP プライマリがパケットを転送するネクストホップ ルータとしてのみ想定されているからです。アプリケーションによっては、Cisco NX-OS が仮想ルータ IP 宛のパケットを受け付けるようにする必要があります。仮想 IP アドレスに secondary オプションを使用すると、ローカル ルータが VRRP マスターの場合、これらのパケットを受け付けるようになります。

VRRP グループを設定した場合は、そのグループをアクティブにするために、グループを明示的に有効にする必要があります。

始める前に

インターフェイス上で IP アドレスを設定していることを確認します。IPv4 アドレス指定の設定（31 ページ）を参照してください。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrp** *number*
4. **address** *ip-address* [**secondary**]
5. **no shutdown**
6. （任意） **show vrrp**
7. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**vrrp** *number*
	例：
	`switch(config-if)# vrrp 250`
	`switch(config-if-vrrp)#`
		仮想ルータ グループを作成します。範囲は 1 ～ 255 です。

Step 4	**address** *ip-address* [**secondary**] 例： `switch(config-if-vrrp)# address 192.0.2.8`	指定の VRRP グループに仮想 IPv4 アドレスを設定します。このアドレスは、インターフェイスの IPv4 アドレスと同じサブネットになければなりません。 **secondary** オプションは、VRRP ルータが仮想ルータの IP アドレスに送信されたパケットを受け付けて、アプリケーションに配信することをアプリケーションが要求する場合に限られます。
Step 5	**no shutdown** 例： `switch(config-if-vrrp)# no shutdown`	VRRP グループを有効にします。デフォルトでは無効になっています。
Step 6	（任意） **show vrrp** 例： `switch(config-if-vrrp)# show vrrp`	VRRP 情報の要約を表示します。
Step 7	（任意） **copy running-config startup-config** 例： `switch(config-if-vrrp)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

VRRP プライオリティの設定

仮想ルータの有効なプライオリティ範囲は 1 ～ 254 です（1 が最下位、254 が最上位のプライオリティ）。バックアップのデフォルトのプライオリティ値は 100 です。インターフェイス アドレスがプライマリ仮想 IP アドレスと同じデバイス（プライマリ）の場合、デフォルト値は 255 です。

vPC 対応のインターフェイスで VRRP を設定する場合は、オプションで vPC トランクにフェールオーバーする時期を制御するしきい値の上限と下限を設定できます。バックアップ ルータのプライオリティが下限のしきい値を下回った場合、VRRPは、すべてのバックアップルータ トラフィックを vPC トランク全体に送信し、プライマリ VRRP ルータを通して転送します。バックアップ VRRP ルータのプライオリティがしきい値の上限を超えるまで、VRRPはこの処理を継続します。

始める前に

インターフェイス上で IP アドレスを設定していることを確認します。IPv4 アドレス指定の設定（31 ページ）を参照してください。

VRRP が有効になっていることを確認します。（「VRRP の設定」の項を参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrp** *number*
4. **shutdown**
5. **priority** *level* [**forwarding-threshold lower** *lower-value* **upper** *upper-value*]
6. **no shutdown**
7. （任意） **show vrrp**
8. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

	コマンドまたはアクション	目的

Step 3	**vrrp** *number*
	例：
	`switch(config-if)# vrrp 250`
	`switch(config-if-vrrp)#`
		仮想ルータ グループを作成します。

Step 4	**shutdown** 例： `switch(config-if-vrrp)# shutdown`	VRRP グループを無効にします。
Step 5	**priority** *level* [**forwarding-threshold lower** *lower-value* **upper** *upper-value*] 例： `switch(config-if-vrrp)# priority 60 forwarding-threshold lower 40 upper 50`	VRRP グループでのアクティブ ルータ選択に使用するプライオリティ レベルを設定します。レベルの範囲は 1 ー 254 です。バックアップの場合、デフォルトは 100 です。インターフェイス IP アドレスが仮想 IP アドレスと等しいプライマリの場合は 255 です。 オプションで、vPC トランクにフェールオーバーする時点を決定するために vPC が使用するしきい値の上限と下限を設定します。*lower-value* の範囲は 1 ～ 255 です。デフォルトは 1 です。*upper-value* の範囲は 1 ～ 255 です。デフォルトは 255 です。
Step 6	**no shutdown** 例： `switch(config-if-vrrp)# no shutdown`	VRRP グループを有効にします。
Step 7	（任意） **show vrrp** 例： `switch(config-if-vrrp)# show vrrp`	VRRP 情報の要約を表示します。
Step 8	（任意） **copy running-config startup-config** 例： `switch(config-if-vrrp)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

VRRP 認証の設定

VRRP グループに単純なテキスト認証を設定できます。

始める前に

インターフェイス上で IP アドレスを設定していることを確認します（IPv4 アドレス指定の設定（31 ページ）を参照）。

VRRP がイネーブルになっていることを確認します（「VRRP の設定」の項を参照）。

ネットワーク上のすべての VRRP デバイスで、認証設定が同じであることを確認します。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrp** *number*
4. **shutdown**
5. **authentication text** *password*
6. **no shutdown**
7. （任意） **show vrrp**
8. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**vrrp** *number*
	例：
	`switch(config-if)# vrrp 250`
	`switch(config-if-vrrp)#`
		仮想ルータ グループを作成します。

Step 4	**shutdown** 例： `switch(config-if-vrrp)# shutdown`	VRRP グループを無効にします。
Step 5	**authentication text** *password* 例： `switch(config-if-vrrp)# authentication text aPassword`	単純なテキスト認証オプションを指定し、キーネーム パスワードを指定します。キーネームの範囲は 1 ～ 255 文字です。16 文字以上を推奨します。テキスト パスワードは、英数字で最大 8 文字です。
Step 6	**no shutdown** 例： `switch(config-if-vrrp)# no shutdown`	VRRP グループを有効にします。デフォルトでは無効になっています。

	コマンドまたはアクション	目的
Step 7	（任意） **show vrrp** 例： `switch(config-if-vrrp)# show vrrp`	VRRP 情報の要約を表示します。
Step 8	（任意） **copy running-config startup-config** 例： `switch(config-if-vrrp)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

アドバタイズメント パケットのタイム インターバルの設定

アドバタイズメント パケットのタイム インターバルを設定できます。

始める前に

インターフェイス上で IP アドレスを設定していることを確認します（IPv4 アドレス指定の設定（31 ページ）を参照）。

VRRP がイネーブルになっていることを確認します（「VRRP の設定」の項を参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrp** *number*
4. **shutdown**
5. **advertisement interval** *seconds*
6. **no shutdown**
7. （任意） **show vrrp**
8. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します

Step 2	**interface** *interface-type slot/port* 例：	インターフェイス設定モードを開始します。

	コマンドまたはアクション	目的
	`switch(config)# interface ethernet 2/1` `switch(config-if)#`	

Step 3	**vrrp** *number*
	例：
	`switch(config-if)# vrrp 250`
	`switch(config-if-vrrp)#`
		仮想ルータ グループを作成します。

Step 4	**shutdown** 例： `switch(config-if-vrrp)# shutdown`	VRRP グループを無効にします。
Step 5	**advertisement interval** *seconds* 例： `switch(config-if-vrrp)# advertisement-interval 15`	アドバタイズメント フレームの送信間隔を秒数で設定します。範囲は 1 ～ 255 です。デフォルト値は 1 秒です。
Step 6	**no shutdown** 例： `switch(config-if-vrrp)# no shutdown`	VRRP グループを有効にします。
Step 7	（任意） **show vrrp** 例： `switch(config-if-vrrp)# show vrrp`	VRRP 情報の要約を表示します。
Step 8	（任意） **copy running-config startup-config** 例： `switch(config-if-vrrp)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

プリエンプションのディセーブル化

VRRP グループ メンバーのプリエンプションをディセーブルにできます。プリエンプションをディセーブルにした場合は、プライオリティのより高いバックアップ ルータが、プライオリティのより低いプライマリ ルータを引き継ぐことはありません。プリエンプションはデフォルトでイネーブルです。

始める前に

インターフェイス上で IP アドレスを設定していることを確認します。IPv4 アドレス指定の設定（31 ページ）を参照してください。

VRRP が有効になっていることを確認します。「VRRPの設定」の項を参照してください。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrp** *number*
4. **shutdown**
5. **no preempt**
6. **no shutdown**
7. （任意） **show vrrp**
8. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**vrrp** *number*
	例：
	`switch(config-if)# vrrp 250`
	`switch(config-if-vrrp)#`
		仮想ルータ グループを作成します。

Step 4	**shutdown** 例： `switch(config-if-vrrp)# shutdown`	VRRP グループを無効にします。
Step 5	**no preempt** 例： `switch(config-if-vrrp)# no preempt`	preempt オプションをディセーブルにして、プライオリティが上位のバックアップが使用されてもプライマリが変わらないようにします。
Step 6	**no shutdown** 例： `switch(config-if-vrrp)# no shutdown`	VRRP グループを有効にします。
Step 7	（任意） **show vrrp** 例：	VRRP 情報の要約を表示します。

	コマンドまたはアクション	目的
	`switch(config-if-vrrp)# show vrrp`	
Step 8	（任意） **copy running-config startup-config** 例： `switch(config-if-vrrp)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

VRRP インターフェイス ステート トラッキングの設定

インターフェイス ステート トラッキングでは、デバイス内の他のインターフェイスのステートに基づいて、仮想ルータのプライオリティが変更されます。トラッキング対象のインターフェイスがダウンしたり、IP アドレスが削除されると、Cisco NX-OS はトラッキング プライオリティ値を仮想ルータに割り当てます。トラッキング対象のインターフェイスがオンライン状態になり、IP アドレスがこのインターフェイスに設定されると、Cisco NX-OS は仮想ルータに設定されていたプライオリティを復元します（「VRRP プライオリティの設定」を参照）。

（注） VRRP はレイヤ 2 インターフェイスのトラッキングをサポートしていません。

始める前に

インターフェイス上で IP アドレスを設定していることを確認します（IPv4 アドレス指定の設定（31 ページ）を参照）。

VRRP がイネーブルになっていることを確認します（「VRRP の設定」の項を参照）。

仮想ルータが有効になっていることを確認します（「VRRP グループの設定」の項を参照）。

インターフェイスでプリエンプションが有効になっていることを確認します。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrp** *number*
4. **shutdown**
5. **track interface** *type slot/port* **priority** *value*
6. **no shutdown**
7. （任意） **show vrrp**
8. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します

Step 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**vrrp** *number*
	例：
	`switch(config-if)# vrrp 250`
	`switch(config-if-vrrp)#`
		仮想ルータ グループを作成します。

Step 4	**shutdown** 例： `switch(config-if-vrrp)# shutdown`	VRRP グループを無効にします。
Step 5	**track interface** *type slot/port* **priority** *value* 例： `switch(config-if-vrrp)# track interface ethernet 2/10 priority 254`	VRRP グループのインターフェイス プライオリティ トラッキングをイネーブルにします。プライオリティの範囲は 1 ～ 254 です。
Step 6	**no shutdown** 例： `switch(config-if-vrrp)# no shutdown`	VRRP グループを有効にします。
Step 7	（任意） **show vrrp** 例： `switch(config-if-vrrp)# show vrrp`	VRRP 情報の要約を表示します。
Step 8	（任意） **copy running-config startup-config** 例： `switch(config-if-vrrp)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

VRRP オブジェクト トラッキングの設定

VRRP を使用して IPv4 オブジェクトを追跡できます。

始める前に

VRRP が有効になっていることを確認します。

「オブジェクト トラッキングの設定」セクションのコマンドを使用して、オブジェクト トラッキングを設定します。

手順の概要

1. **configure terminal**
2. **interface type number**
3. **vrrp** *number* **address-family ipv4**
4. **track** *object-number* **decrement** *number*
5. （任意） **show running-config vrrp**
6. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**interface type number**
	例：
	`switch(config)#`
	`switch(config-if)# interface ethernet 2/1`
	`switch(config-if)#`
		インターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。

Step 3	**vrrp** *number* **address-family ipv4**
	例：
	`switch(config-if)# vrrp 5`
	`address-family ipv4`
	`switch(config-if-vrrp-group)#`
		IPv4 用に VRRP グループを作成し、VRRP vrrp number address-family ipv4 グループ設定モードを開始します。範囲は 1 ～ 255 です。

Step 4	**track** *object-number* **decrement** *number*
	例：
	`switch(config-if-vrrp-group)# track 1`
	`decrement 2`
		仮想ルータ グループを作成します。範囲は 1 ～ 255 です。

	コマンドまたはアクション	目的
Step 5	（任意） **show running-config vrrp** 例： `switch(config-if-vrrp-group)# show running-config vrrp`	VRRP の実行中の設定を表示します。
Step 6	（任意） **copy running-config startup-config** 例： `switch(config-if-vrrp-group)# copy running-config startup-config`	この設定変更を保存します。

VRRPv3 の設定

VRRPv3 および VRRS の有効化

VRRPv3 グループを設定して有効にするには、その前に VRRPv3 をグローバルで有効にする必要があります。

手順の概要

1. **configure terminal**
2. [**no**] **feature vrrpv3**
3. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	[**no**] **feature vrrpv3** 例： `switch(config)# feature vrrpv3`	VRRP バージョン 3 と仮想ルータ冗長サービス（VRRS）をイネーブルにします。このコマンドの **no** 形式を使用すると、VRRPv3 および VRRS が無効になります。 VRRPv2 が現在設定されている場合は、グローバル設定モードで **no feature vrrp** コマンドを使用して

	コマンドまたはアクション	目的
		VRRPv2 設定を削除し、その後 **feature vrrpv3** コマンドを使用して VRRPv3 を有効にします。
Step 3	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

VRRPv3 グループの作成

VRRPv3 グループを作成し、仮想 IP アドレスを割り当て、グループをイネーブルにすることができます。

始める前に

VRRPv3 が有効になっていることを確認します。

インターフェイスに IP アドレスが設定されていることを確認します。

手順の概要

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **vrrpv3** *number* **address-family** [**ipv4** | **ipv6**]
4. （任意） **address** *ip-address* [**primary** | **secondary**]
5. （任意） **description** 説明
6. （任意） **match-address**
7. （任意） **preempt** [**delay minimum** *seconds*]
8. （任意） **priority** *level*
9. （任意） **timers advertise** *interval*
10. （任意） **vrrp2**
11. （任意） **vrrs leader** *vrrs-leader-name*
12. （任意） **shutdown**
13. （任意） **show fhrp** [*interface-type interface-number*] [**verbose**]
14. （任意） **show vrrpv3** *interface-type interface-number*
15. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**interface ethernet** *slot/port*
	例：
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**vrrpv3** *number* **address-family** [**ipv4** | **ipv6**]
	例：
	`switch(config-if)# vrrpv3 5 address-family ipv4`
	`switch(config-if-vrrpv3-group)#`
		VRRPv3 グループを作成し、VRRPv3 グループ設定モードを開始します。範囲は 1 ～ 255 です。

Step 4	（任意） **address** *ip-address* [**primary** | **secondary**] 例： `switch(config-if-vrrpv3-group)# address 100.0.1.10 primary`	VRRPv3 グループのプライマリ アドレスまたはセカンダリ IPv4 または IPv6アドレスを指定します。 VRRPv3 グループでセカンダリ IP アドレスを使用するには、まず同じグループでプライマリ IP アドレスを設定する必要があります。
Step 5	（任意） **description** 説明 例： `switch(config-if-vrrpv3-group)# description group3`	VRRPv3 グループの説明を指定します。最大 80 文字の英数字を入力できます。
Step 6	（任意） **match-address** 例： `switch(config-if-vrrpv3-group)# match-address`	アドバタイズメント パケットのセカンダリ アドレスを設定したアドレスと照合します。
Step 7	（任意） **preempt** [**delay minimum** *seconds*] 例： `switch(config-if-vrrpv3-group)# preempt delay minimum 30`	オプションの延期時間を指定して、プライオリティの低いプライマリ スイッチのプリエンプションをイネーブルにします。範囲は 0〜3600 です。
Step 8	（任意） **priority** *level* 例： `switch(config-if-vrrpv3-group)# priority 3`	VRRPv3 グループのプライオリティを指定します。範囲は 1〜254 です。

	コマンドまたはアクション	目的
Step 9	（任意） **timers advertise** *interval* 例： `switch(config-if-vrrpv3-group)# timers advertise 1000`	アドバタイズメントタイマーを設定します（ミリ秒単位）。範囲は 100ー40950 です。 シスコは、このタイマーを 1 秒以上の値に設定することを推奨します。
Step 10	（任意） **vrrp2** 例： `switch(config-if-vrrpv3-group)# vrrp2`	VRRPv2 のみをサポートしているデバイスとの相互運用性を確保するために、VRRPv2 に対するサポートも同時にイネーブルにします。 VRRPv2 互換モードは、VRRPv2 から VRRPv3 にアップグレードするために提供されます。これは完全な VRRPv2 実装ではないので、アップグレードを実行する場合にのみ使用してください。
Step 11	（任意） **vrrs leader** *vrrs-leader-name* 例： `switch(config-if-vrrpv3-group)# vrrs leader leader1`	VRRS に登録するリーダーの名前を指定します。
Step 12	（任意） **shutdown** 例： `switch(config-if-vrrpv3-group)# shutdown`	VRRPv3 グループの VRRP 設定を無効にします。
Step 13	（任意） **show fhrp** [*interface-type interface-number*] [**verbose**] 例： `switch(config-if-vrrpv3-group)# show fhrp ethernet 2/1 verbose`	ファースト ホップ冗長性プロトコル（FHRP）の情報を表示します。詳細情報を表示するには、**verbose** キーワードを使用します。
Step 14	（任意） **show vrrpv3** *interface-type interface-number* 例： `switch(config-if-vrrpv3-group)# show vrrpv3 ethernet 2/1`	指定されたインターフェイスに関する VRRPv3 設定情報を表示します。
Step 15	（任意） **copy running-config startup-config** 例： `switch(config-if-vrrpv3-group)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップコンフィギュレーションにコピーします。

VRRPv3 コントロールグループの設定

VRRPv3 コントロール グループを設定できます。

始める前に

VRRPv3 が有効になっていることを確認します。

インターフェイスに IP アドレスが設定されていることを確認します。

手順の概要

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **ip address** *ip-address mask* [**secondary**]
4. **vrrpv3** *number* **address-family** [**ipv4** | **ipv6**]
5. （任意） **address** *ip-address* [**primary** | **secondary**]
6. （任意） **shutdown**
7. （任意） **show fhrp** [*interface-type interface-number*] [**verbose**]
8. （任意） **show vrrpv3** *interface-type interface-number*
9. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**interface ethernet** *slot/port*
	例：
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**ip address** *ip-address mask* [**secondary**] 例： `switch(config-if)# ip address 209.165.200.230 255.255.255.224`	インターフェイスの IP アドレスを設定します。 **secondary** キーワードを使用して、インターフェイスで追加の IP アドレスを設定できます。

Step 4	**vrrpv3** *number* **address-family** [**ipv4** | **ipv6**]
	例：
	`switch(config-if)# vrrpv3 5 address-family ipv4`
	`switch(config-if-vrrpv3-group)#`
		VRRPv3 グループを作成し、VRRPv3 グループ設定モードを開始します。範囲は 1 ～ 255 です。

Step 5	（任意） **address** *ip-address* [**primary** | **secondary**] 例：	VRRPv3 グループのプライマリ アドレスまたはセカンダリ IPv4 または IPv6アドレスを指定します。

	コマンドまたはアクション	目的
	`switch(config-if-vrrpv3-group)# address 209.165.200.227 primary`	
Step 6	（任意） **shutdown** 例： `switch(config-if-vrrpv3-group)# shutdown`	VRRPv3 グループの VRRP 設定を無効にします。
Step 7	（任意） **show fhrp** [*interface-type interface-number*] [**verbose**] 例： `switch(config-if-vrrpv3-group)# show fhrp ethernet 2/1 verbose`	ファーストホップ冗長性プロトコル（FHRP）の情報を表示します。詳細情報を表示するには、**verbose** キーワードを使用します。
Step 8	（任意） **show vrrpv3** *interface-type interface-number* 例： `switch(config-if-vrrpv3-group)# show vrrpv3 ethernet 2/1`	指定されたインターフェイスに関する VRRPv3 設定情報を表示します。
Step 9	（任意） **copy running-config startup-config** 例： `switch(config-if-vrrpv3-group)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

VRRPv3 オブジェクト トラッキングの設定

VRRPv3 を使用して IPv4 または IPv4 オブジェクトを追跡できます。

始める前に

VRRPv3 が有効になっていることを確認します。

「オブジェクトトラッキングの設定」セクションのコマンドを使用して、オブジェクトトラッキングを設定します。

手順の概要

1. **configure terminal**
2. **interface type number**
3. **vrrpv3** *number* **address-family [ipv4 | ipv6]**
4. **track** *object-number* **decrement** *number*
5. （任意） **show running-config vrrpv3**
6. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**interface type number**
	例：
	`switch(config)#`
	`switch(config-if)# interface ethernet 2/1`
	`switch(config-if)#`
		インターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。

Step 3	**vrrpv3** *number* **address-family [ipv4 | ipv6]**
	例：
	`switch(config-if)# vrrpv3 5`
	`address-family ipv6`
	`switch(config-if-vrrpv3-group)#`
		IPv4 または IPv6 に対して VRRPv3 グループを作成し、VRRPv3 グループ設定モードを開始します。範囲は 1 ～ 255 です。

Step 4	**track** *object-number* **decrement** *number*
	例：
	`switch(config-if-vrrpv3-group)# object-track 1`
	`decrement 2`
		VRRPv3 グループを使用して IPv6 オブジェクトのステートを追跡するようにトラッキング プロセスを設定します。インターフェイスの VRRPv3 は、VRRPv3 グループでオブジェクトに何らかの変更が生じた場合には通知されるように、トラッキング プロセスに登録します。インターフェイスの IPv6 オブジェクト ステートがダウンになると、VRRPv3 グループのプライオリティは、指定された数値だけ引き下げられます。

Step 5	（任意） **show running-config vrrpv3**
	例：
	`switch(config-if-vrrp-group)# show`
	`running-config vrrp`
		VRRP の実行中の設定を表示します。

Step 6	（任意） **copy running-config startup-config**
	例：
	`switch(config-if-vrrp-group)# copy`
	`running-config startup-config`
		この設定変更を保存します。

VRRS 経路の設定

仮想ルータ冗長サービス（VRRS）の経路を設定できます。拡張環境では、VRRS 経路は VRRPv3 制御グループと組み合わせて使用する必要があります。

始める前に

VRRPv3 が有効になっていることを確認します。

インターフェイスに IP アドレスが設定されていることを確認します。

手順の概要

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **ip address** *ip-address mask* [**secondary**]
4. **vrrs pathway** *vrrs-tag*
5. **mac address** {*mac-address* | **inherit**}
6. **address** *ip-address*
7. （任意） **show vrrs pathway** *interface-type interface-number*
8. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**interface ethernet** *slot/port*
	例：
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

Step 3	**ip address** *ip-address mask* [**secondary**] 例： `switch(config-if)# ip address 209.165.200.230 255.255.255.224`	インターフェイスの IP アドレスを設定します。 **secondary** キーワードを使用して、インターフェイスで追加の IP アドレスを設定できます。

Step 4	**vrrs pathway** *vrrs-tag*
	例：
	`switch(config-if)# vrrs pathway path1`
	`switch(config-if-vrrs-pw)#`
		VRRS グループの VRRS 経路を定義し、VRRS 経路コンフィギュレーション モードを開始します。
		vrrs-tag 引数は、経路に関連付けられている VRRS タグの名前を指定します。

	コマンドまたはアクション	目的
Step 5	**mac address** {*mac-address* | **inherit**} 例： `switch(config-if-vrrs-pw)# mac address fe24.fe24.fe24`	経路の MAC アドレスを指定します。 **inherit** キーワードを使用すると、経路は関連付けられている VRRPv3 グループの仮想 MAC アドレスを継承します。
Step 6	**address** *ip-address* 例： `switch(config-if-vrrs-pw)# address 209.165.201.10`	経路の仮想 IPv4 アドレスまたは IPv6 アドレスを定義します。 VRRPv3 グループは、複数の経路を制御できます。
Step 7	（任意） **show vrrs pathway** *interface-type interface-number* 例： `switch(config-if-vrrs-pw)# show vrrs pathway ethernet 1/2`	異なる経路の状態（アクティブ、非アクティブ、非対応など）に関する VRRS 経路の情報を表示します。
Step 8	（任意） **copy running-config startup-config** 例： `switch(config-if-vrrs-pw)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

VRRP の設定の確認

VRRP 設定情報を表示するには、次のいずれかの作業を実行します。

コマンド	目的
show interface *interface-type*	インターフェイスの仮想ルータ設定を表示します。
show fhrp *interface-type interface-number*	ファースト ホップ冗長性プロトコル（FHRP）の情報を表示します。
show vrrp [*group-number*]	すべてのグループまたは特定の VRRP グループについて、VRRP ステータスを表示します。

VRRPv3 設定の確認

VRRPv3 の設定 の設定情報を表示するには、次のいずれかの作業を行います。

コマンド	目的
show vrrpv3[**all** |**brief** |**detail**]	VRRPv3 の設定情報を表示します。

コマンド	目的
show vrrpv3 *interface-type interface-number*	特定のインターフェイスに関する VRRPv3 設定情報を表示します。
show vrrs client [*client-name*]	VRRS クライアント情報を表示します。
show vrrs pathway [*interface-type interface-number*]	異なる経路の状態（アクティブ、非アクティブ、非対応など）に関する VRRS 経路の情報を表示します。
show vrrs server	VRRS サーバ情報を表示します。
show vrrs tag [*tag-name*]	VRRS タグ情報を表示します。

VRRP 統計情報のモニタリングとクリア

VRRP の統計情報を表示するには、次のコマンドを使用します。

コマンド	目的
show vrrp statistics	VRRP の統計情報を表示します。

デバイスのすべてのインターフェイスについて、すべての VRRP 統計情報を消去するには、**clear vrrp statistics** コマンドを使用します。

VRRPv3 統計情報のモニタリングとクリア

VRRPv3 統計情報を表示するには、次のコマンドを使用します。

コマンド	目的
show vrrpv3 statistics	VRRPv3 統計情報を表示します。

clear vrrpv3 statistics を使用します コマンドを使用して、デバイスのすべてのインターフェイスについて、VRRPv3 統計情報をクリアします。

VRRP の設定例

この例では、ルータ A とルータ B はそれぞれ 3 つの VRRP グループに属しています。コンフィギュレーションにおいて、各グループのプロパティは次のとおりです。

- グループ 1：
  - 仮想 IP アドレスは 10.1.0.10 です。

- ルータ A は優先順位 120 で、このグループのプライマリになります。
- アドバタイズ インターバルは 3 秒です。
- プリエンプションは有効です。

- グループ 5：
  - ルータ B はプライオリティ 200 で、このグループのマスターになります。
  - アドバタイズ インターバルは 30 秒です。
  - プリエンプションは有効です。

- グループ 100：
  - ルータ A は、IP アドレスが上位（10.1.0.2）なので、このグループのプライマリになります。
  - アドバタイズ インターバルはデフォルトの 1 秒です。
  - プリエンプションは無効です。

ルータ A

```
switch (config)# interface ethernet 1/1
switch (config-if)# ip address 10.1.0.1/16
switch (config-if)# no shutdown
switch (config-if)# vrrp 1
switch (config-if-vrrp)# priority 120
switch (config-if-vrrp)# authentication text cisco
switch (config-if-vrrp)# advertisement-interval 3
switch (config-if-vrrp)# address 10.1.0.10
switch (config-if-vrrp)# no shutdown
switch (config-if-vrrp)# exit
switch (config-if)# vrrp 5
switch (config-if-vrrp)# priority 100
switch (config-if-vrrp)# advertisement-interval 30
switch (config-if-vrrp)# address 10.1.0.50
switch (config-if-vrrp)# no shutdown
switch (config-if-vrrp)# exit
switch (config-if)# vrrp 100
switch (config-if-vrrp)# no preempt
switch (config-if-vrrp)# address 10.1.0.100
switch (config-if-vrrp)# no shutdown
```

ルータ B

```
switch (config)# interface ethernet 1/1
switch (config-if)# ip address 10.1.0.2/16
switch (config-if)# no shutdown
switch (config-if)# vrrp 1
switch (config-if-vrrp)# priority 100
switch (config-if-vrrp)# authentication text cisco
switch (config-if-vrrp)# advertisement-interval 3
switch (config-if-vrrp)# address 10.1.0.10
switch (config-if-vrrp)# no shutdown
switch (config-if-vrrp)# exit
switch (config-if)# vrrp 5
```

```
switch (config-if-vrrp)# priority 200
switch (config-if-vrrp)# advertisement-interval 30
switch (config-if-vrrp)# address 10.2.0.50
switch (config-if-vrrp)# no shutdown
switch (config-if-vrrp)# exit
switch (config-if)# vrrp 100
switch (config-if-vrrp)# no preempt
switch (config-if-vrrp)# address 10.2.0.100
switch (config-if-vrrp)# no shutdown
```

VRRPv3 の設定例

次に、VRRPv3 をイネーブルにし VRRPv3 グループを作成およびカスタマイズする例を示します。

```
switch# configure terminal
switch(config)# feature vrrpv3
switch(config)# interface ethernet 4/6
switch(config-if)# vrrpv3 5 address-family ipv4
switch(config-if-vrrp3-group)# address 209.165.200.225 primary
switch(config-if-vrrp3-group)# description group3
switch(config-if-vrrp3-group)# match-address
switch(config-if-vrrp3-group)# preempt delay minimum 30
switch(config-if-vrrpv3-group)# show fhrp ethernet 4/6 verbose
switch(config-if-vrrpv3-group)# show vrrpv3 ethernet 4/6
```

次に、VRRPv3 制御グループを設定する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ip address 209.165.200.230 255.255.255.224
switch(config-if)# vrrpv3 5 address-family ipv4
switch(config-if-vrrpv3-group)# address 209.165.200.227 primary
switch(config-if-vrrpv3-group)# vrrs leader leader1
switch(config-if-vrrpv3-group)# shutdown
switch(config-if-vrrpv3-group)# show fhrp ethernet 1/2 verbose
switch(config-if-vrrpv3-group)# show vrrpv3 ethernet 1/2
```

次に、VRRPv3 のオブジェクト トラッキングを設定する例を示します。

```
track 1 interface Ethernet1/12 ip routing
track 2 interface Ethernet1/12 ipv6 routing
track 3 interface Ethernet1/12 line-protocol
track 4 interface Ethernet1/12.1 ip routing
track 5 interface Ethernet1/12.1 ipv6 routing
track 6 interface Ethernet1/12.1 line-protocol
track 7 interface loopback1 ip routing
track 8 interface loopback1 ipv6 routing
track 9 interface loopback1 line-protocol
track 10 interface port-channel1 ip routing
track 11 interface port-channel1 ipv6 routing
track 12 interface port-channel1 line-protocol
track 13 ip route 170.10.10.10/24 reachability
track 14 ip route 180.10.10.0/24 reachability hmm
track 15 ipv6 route 2001::170:10:10:10/128 reachability
track 16 list boolean and
object 1
object 2
interface Vlan10
vrrpv3 10 address-family ipv4
```

```
timers advertise 100
priority 200
object-track 1 decrement 2
object-track 2 decrement 2
object-track 3 decrement 2
object-track 4 decrement 2
object-track 5 decrement 2
object-track 6 decrement 2
object-track 7 decrement 2
object-track 8 decrement 2
object-track 9 decrement 2
object-track 10 decrement 2
address 10.10.10.3 primary
interface Vlan10
vrrpv3 10 address-family ipv6
timers advertise 100
priority 200
object-track 1 decrement 4
object-track 2 decrement 4
object-track 3 decrement 4
object-track 4 decrement 4
object-track 5 decrement 4
object-track 6 decrement 4
object-track 7 decrement 4
object-track 8 decrement 4
```

次に、VRRS 経路を設定する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ip address 209.165.200.230 255.255.255.224
switch(config-if)# vrrs pathway path1
switch(config-if-vrrs-pw)# mac address inherit
switch(config-if-vrrs-pw)# address 209.165.201.10
switch(config-if-vrrs-pw)# show vrrs pathway ethernet 1/2
```

その他の参考資料

VRRP の関連資料

関連項目	マニュアル タイトル
Hot Standby Router Protocol（HSRP）の設定	『Configuring HSRP』（651 ページ）
高可用性の設定	『Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide』

第 21 章

オブジェクト トラッキングの設定

この章は、次の項で構成されています。

- オブジェクト トラッキングについて （717 ページ）
- オブジェクト トラッキングの設定例 （719 ページ）
- オブジェクト トラッキングに関する注意事項と制約事項 （719 ページ）
- デフォルト設定 （720 ページ）
- オブジェクト トラッキングの設定 （720 ページ）
- オブジェクト トラッキングの設定の確認 （732 ページ）
- オブジェクト トラッキングの設定例 （732 ページ）
- 関連項目 （733 ページ）
- その他の参考資料 （733 ページ）

オブジェクト トラッキングについて

オブジェクトトラッキングを使用すると、インターフェイスラインプロトコルステート、IP ルーティング、ルート到達可能性などの、デバイス上の特定のオブジェクトをトラッキングし、トラッキング対象オブジェクトのステートが変化したときに対処できます。この機能により、ネットワークのアベイラビリティが向上し、オブジェクトがダウンした場合のリカバリ時間が短縮されます。

オブジェクト トラッキングの概要

オブジェクトトラッキングを使用すると、インターフェイスラインプロトコルステート、IP ルーティング、ルート到達可能性などの、デバイス上の特定のオブジェクトをトラッキングし、トラッキング対象オブジェクトのステートが変化したときに対処できます。この機能により、ネットワークのアベイラビリティが向上し、オブジェクトがダウンした場合のリカバリ時間が短縮されます。

オブジェクト トラッキング機能を使用すると、トラッキング対象オブジェクトを作成できます。複数のクライアントでこのオブジェクトを使用し、トラッキング対象オブジェクトが変化したときのクライアント動作を変更できます。複数のクライアントがそれぞれの関心をトラッキング プロセスに登録し、同じオブジェクトをトラッキングし、オブジェクトのステートが変化したときに異なるアクションを実行します。

クライアントには次の機能が含まれます。

- Embedded Event Manager（EEM）
- ホットスタンバイ冗長プロトコル（HSRP）
- 仮想ポート チャネル（vPC）
- 仮想ルータ冗長プロトコル（VRRP）および VRRPv3

オブジェクト トラッキングは、トラッキング対象オブジェクトのステータスをモニタし、変更があった場合は関係クライアントに伝えます。各トラッキング対象オブジェクトは、一意の番号で識別します。クライアントはこの番号を使用して、トラッキング対象オブジェクトのステートが変化したときに実行するアクションを設定できます。

Cisco NX-OS がトラッキングするオブジェクト タイプは、次のとおりです。

- インターフェイス ライン プロトコル ステート：ライン プロトコル ステートがアップまたはダウンかどうかをトラッキングします。
- インターフェイス IP ルーティング ステート：インターフェイスに IPv4 または IPv6 アドレスが設定されていて、IPv4 または IPv6 ルーティングが有効でアクティブかどうかをトラッキングします。
- IP ルート到達可能性：IPv4 または IPv6 ルートが存在していて、ローカル デバイスから到達可能かどうかをトラッキングします。

たとえば、HSRP を設定すると、冗長ルータの 1 つをネットワークの他の部分に接続するインターフェイスのラインプロトコルをトラッキングできます。リンクプロトコルがダウンした場合、影響を受ける HSRP ルータのプライオリティを変更し、よりすぐれたネットワーク接続が得られるバックアップ ルータにスイッチオーバーされるようにできます。

オブジェクト トラッキング リスト

オブジェクト トラッキング リストを使用すると、複数のオブジェクトのステートをまとめてトラッキングできます。オブジェクト トラッキング リストは次の機能をサポートします。

- ブール「and」機能：トラッキング リスト オブジェクトがアップになるには、トラッキング リスト内に定義された各オブジェクトがアップ状態である必要があります。
- ブール「or」機能：トラッキング対象オブジェクトがアップになるには、トラッキング リスト内に定義された少なくとも 1 つのオブジェクトがアップ状態である必要があります。
- しきい値パーセンテージ：トラッキング対象リストに含まれるアップ オブジェクトのパーセンテージが、アップ状態になるトラッキング リストの設定されたアップしきい値を上回っている必要があります。トラッキング対象リストに含まれるダウン オブジェクトのパーセンテージが設定されたトラッキング リストのダウンしきい値を上回っている場合、トラッキング対象リストはダウンとしてマークされます。
- しきい値の重み：トラッキング対象リスト内の各オブジェクトに重み値を割り当て、トラッキング リストに重みしきい値を割り当てます。すべてのアップ オブジェクトの重み値の合計

がトラッキング リストの重みアップしきい値を超えている場合、トラッキング リストはアップ状態になります。すべてのダウン オブジェクトの重み値の合計がトラッキング リストの重みダウンしきい値を超えている場合、トラッキング リストはダウン状態になります。

他のエンティティ（たとえば、仮想ポート チャネル（vPC））は、オブジェクト トラッキング リストを使用することにより、vPC を作成する複数のピア リンクのステートに基づいて vPC のステートを変更できます。vPC の詳細については、『Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide』を参照してください。

トラックリストの詳細については、「ブール式によるオブジェクトトラックリストの設定」を参照してください。

高可用性

オブジェクト トラッキングは、ステートフル リスタートを通じてハイ アベイラビリティをサポートします。ステートフル リスタートが実行されるのは、オブジェクト トラッキング プロセスがクラッシュした場合です。オブジェクト トラッキングは、デュアル スーパーバイザ システムでのステートフル スイッチオーバーもサポートします。スイッチオーバー後に Cisco NX-OS が実行コンフィギュレーションを適用します。

オブジェクト トラッキングを使用して、ネットワーク全体の可用性が向上するように、クライアントの動作を変更することもできます。

仮想化のサポート

オブジェクト トラッキングは仮想ルーティングおよび転送（VRF）インスタンスをサポートします。Cisco NX-OS はデフォルトで、デフォルト VRF のオブジェクトのルート到達可能ステートをトラッキングします。別の VRF のオブジェクトをトラッキングする場合は、オブジェクトをその VRF のメンバとして設定する必要があります（「デフォルト以外の VRF に対してオブジェクト トラッキングを設定する」の項を参照）。

オブジェクト トラッキングの設定例

次の例は、ルート到達可能性に対してオブジェクト トラッキングを設定し、VRF Red を使用してルートの到達可能性情報を調べる方法を示しています。

```
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
```

オブジェクト トラッキングに関する注意事項と制約事項

オブジェクト トラッキング設定時の注意事項および制約事項は、次のとおりです。

- イーサネット、サブインターフェイス、ポートチャネル、ループバックインターフェイス、および VLAN インターフェイスをサポートします。
- HSRP グループごとに 1 つのトラッキング対象オブジェクトをサポートします。
- VRRP および VRRPv3 はオブジェクト トラッキングをサポートします。VRRP および設定の詳細については、「VRRP の設定」を参照してください。

デフォルト設定

次の表に、オブジェクト トラッキング パラメータのデフォルト設定を示します。

表 *33*: デフォルトのオブジェクト トラッキング パラメータ

パラメータ	デフォルト
Tracked object VRF	デフォルト VRF のメンバ

オブジェクト トラッキングの設定

IP SLA オブジェクト トラッキングの設定の詳細については、『Cisco Nexus 9000 Series NX-OS IP SLAs Configuration Guide』を参照してください。

インターフェイスに対するオブジェクト トラッキングの設定

インターフェイスのライン プロトコルまたは IPv4 や IPv6 ルーティングのステートをトラッキングするように Cisco NX-OS を設定できます。

手順の概要

1. **configure terminal**
2. **track** *object-id* **interface** *interface-type number* {**ip routing** | **ipv6 routing** | **line-protocol**}
3. （任意） **show track** [*object-id*]
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
Step 1	**configure terminal** 例:	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	`switch# configure terminal` `switch(config)#`	

Step 2	**track** *object-id* **interface** *interface-type number* {**ip routing** | **ipv6 routing** | **line-protocol**}
	例：
	`switch(config)# track 1 interface ethernet 1/2 line-protocol`
	`switch(config-track)#`
		インターフェイスのトラッキング対象オブジェクトを作成し、トラッキング コンフィギュレーション モードを開始します。*object-id* の範囲は 1 ～ 512 です。

Step 3	（任意） **show track** [*object-id*] 例： `switch(config-track)# show track 1`	オブジェクトのトラッキング情報を表示します。
Step 4	（任意） **copy running-config startup-config** 例： `switch(config-track)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

Ethernet 1/2 上でライン プロトコル ステートのオブジェクト トラッキングを設定する例を示します。

```
switch# configure terminal
switch(config)# track 1 interface ethernet 1/2 line-protocol
switch(config-track)# copy running-config startup-config
```

Ethernet 1/2 上で IPv4 ルーティング ステートのオブジェクト トラッキングを設定する例を示します。

```
sswitch# configure terminal
switch(config)# track 2 interface ethernet 1/2 ip routing
switch(config-track)# copy running-config startup-config
```

Ethernet 1/2 上で IPv6 ルーティング ステートのオブジェクト トラッキングを設定する例を示します。

```
switch# configure terminal
switch(config)# track 3 interface ethernet 1/2 ipv6 routing
switch(config-track)# copy running-config startup-config
```

トラッキング オブジェクトの削除

手順の概要

1. **configure terminal**
2. **no track** *object-id*
3. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**no track** *object-id*
	例：
	`switch(config)# no track 1`
	`switch(config-track)#`
		インターフェイスのトラッキング対象オブジェクトを削除します。*object-id* の範囲は 1 ～ 512 です。

Step 3	（任意） **copy running-config startup-config** 例： `switch(config-track)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、トラッキング対象オブジェクトを削除する例を示します。

```
switch# configure terminal
switch(config)# no track 1
switch(config-track)# copy running-config startup-config
```

ルート到達可能性に対するオブジェクト トラッキングの設定

Cisco NX-OSを IP ルートまたは IPv6 ルートの存在と到達可能性をトラッキングするように設定できます。

手順の概要

1. **configure terminal**
2. **track** *object-id* {**ip** | **ipv6**} **route** *prefix*/*length* **reachability**
3. （任意） **show track** [*object-id*]
4. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**track** *object-id* {**ip** | **ipv6**} **route** *prefix*/*length* **reachability**
	例：
	`switch(config)# track 3 ipv6 route 2::5/64 reachability`
	`switch(config-track)#`
		ルートのトラッキング対象オブジェクトを作成し、トラッキング コンフィギュレーション モードを開始します。*object-id* の範囲は 1 ～ 512 です。IPv4 のプレフィックス フォーマットは A.B.C.D/length です。length の範囲は 1 ～ 32 です。IPv6 のプレフィックス フォーマットは A:B::C:D/length です。length の範囲は 1 ～ 128 です。

Step 3	（任意） **show track** [*object-id*] 例： `switch(config-track)# show track 1`	オブジェクトのトラッキング情報を表示します。
Step 4	（任意） **copy running-config startup-config** 例： `switch(config-track)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、デフォルト VRF で IPv4 ルートのオブジェクト トラッキングを設定する例を示します。

```
switch# configure terminal
switch(config)# track 4 ip route 192.0.2.0/8 reachability
switch(config-track)# copy running-config startup-config
```

次に、デフォルト VRF で IPv6 ルートのオブジェクト トラッキングを設定する例を示します。

```
switch# configure terminal
switch(config)# track 5 ipv6 route 10::10/128 reachability
switch(config-track)# copy running-config startup-config
```

ブール式を含むオブジェクト トラッキング リストの設定

複数のトラッキング対象オブジェクトを含むオブジェクト トラッキング リストを設定できます。トラッキング対象リストには1つまたは複数のオブジェクトが含まれます。ブール式では、「and」

または「or」演算子を使用して 2 種類の演算を実行できます。たとえば、「and」演算子を使用して2つのインターフェイスをトラッキングする場合、「アップ」は両方のインターフェイスがアップであることを意味し、「ダウン」はどちらかのインターフェイスがダウンであることを意味します。

手順の概要

1. **configure terminal**
2. **track** *track-number* **list boolean** {**and** | **or**}
3. **object** *object-number* [**not**]
4. （任意） **show track** [*object-id*]
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

Step 2	**track** *track-number* **list boolean** {**and** | **or**}
	例：
	`switch(config)# track 1 list boolean and`
	`switch(config-track)#`
		トラッキング対象リスト オブジェクトを設定し、トラッキング設定モードを開始します。トラッキング対象リストのステートがブール式に基づいて決まることを指定します。キーワードは次のとおりです。
		• and：すべてのオブジェクトがアップである場合にリストがアップになり、1つ以上のオブジェクトがダウンの場合にリストがダウンになることを指定します。たとえば 2 つのインターフェイスをトラッキングする場合、アップは両方のインターフェイスがアップ状態であることを表し、ダウンはいずれかのインターフェイスがダウン状態であることを表します。
		• or：少なくとも 1 つのオブジェクトが稼働している場合、リストが稼働していることを示します。たとえば 2 つのインターフェイスをトラッキングする場合、アップはいずれか一方のインターフェイスがアップ状態であることを意味し、ダウンは両方のインターフェイスがダウン状態であることを意味します。
		track-number の範囲は 1 ～ 512 です。

	コマンドまたはアクション	目的
Step 3	**object** *object-number* [**not**] 例： `switch(config-track)# object 10`	トラッキング リストにトラッキング対象オブジェクトを追加します。*object-id* の範囲は 1 ～ 512 です。オプションの **not** キーワードを指定すると、トラッキング対象オブジェクトのステートが否定されます。 （注） 例では、オブジェクト10がアップのときに、トラッキング対象リストがオブジェクト 10 をダウンとして検出します。
Step 4	（任意） **show track** [*object-id*] 例： `switch(config-track)# show track`	オブジェクトのトラッキング情報を表示します。
Step 5	（任意） **copy running-config startup-config** 例： `switch(config-track)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、複数のオブジェクトを含むトラッキング リストをブール「and」で設定する例を示します。

```
switch# configure terminal
switch(config)# track 1 list boolean and
switch(config-track)# object 10
switch(config-track)# object 20 not
```

パーセンテージしきい値を含むオブジェクト トラッキング リストの設定

パーセンテージしきい値を含むオブジェクトトラッキングリストを設定できます。トラッキング対象リストには 1 つまたは複数のオブジェクトが含まれます。トラッキング リストがアップ状態になるには、アップオブジェクトのパーセンテージがトラッキングリストに設定されたパーセントしきい値を超えている必要があります。たとえば、トラッキング対象リストに 3 つのオブジェクトが含まれており、アップしきい値を 60 % に設定した場合は、2 つのオブジェクト（全オブジェクトの 66 %）がアップ状態になるまで、トラッキング リストがアップ状態になりません。

手順の概要

1. **configure terminal**
2. **track** *track-number* **list threshold percentage**
3. **threshold percentage up** *up-value* **down** *down-value*

4. **object** *object-id*
5. （任意） **show track** [*object-id*]
6. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**track** *track-number* **list threshold percentage**
	例：
	`switch(config)# track 1 list threshold percentage`
	`switch(config-track)#`
		トラッキング対象リスト オブジェクトを設定し、トラッキング設定モードを開始します。トラッキング対象リストのステートが設定されたしきい値パーセントに基づいて決まることを指定します。
		track-number の範囲は 1 ～ 512 です。

Step 3	**threshold percentage up** *up-value* **down** *down-value* 例： `switch(config-track)# threshold percentage up 70 down 30`	トラッキング対象リストのしきい値パーセントを設定します。有効値は 0 ～ 100 パーセントです。
Step 4	**object** *object-id* 例： `switch(config-track)# object 10`	トラッキング リストにトラッキング対象オブジェクトを追加します。*object-id* の範囲は 1 ～ 512 です。
Step 5	（任意） **show track** [*object-id*] 例： `switch(config-track)# show track`	オブジェクトのトラッキング情報を表示します。
Step 6	（任意） **copy running-config startup-config** 例： `switch(config-track)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、アップしきい値が 70 % でダウンしきい値が 30 % の追跡リストを設定する例を示します。

```
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
```

重みしきい値を含むオブジェクト トラッキング リストの設定

重みしきい値を含むオブジェクトトラッキングリストを設定できます。トラッキング対象リストには1つまたは複数のオブジェクトが含まれます。トラッキング リストがアップ ステートになるには、アップ オブジェクトの重み値の合計がトラッキング リストに設定されたアップ重みしきい値を超えている必要があります。たとえば、トラッキング対象リストに重み値がデフォルトの 10 である 3 つのオブジェクトがあり、アップしきい値を 15 に設定した場合、トラッキング リストがアップ状態になるには、2 つのオブジェクトがアップ状態になる（重み値の合計が 20 になる）必要があります。

手順の概要

1. **configure terminal**
2. **track** *track-number* **list threshold weight**
3. **threshold weight up** *up-value* **down** *down-value*
4. **object** *object-id* **weight** *value*
5. （任意） **show track** [*object-id*]
6. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**track** *track-number* **list threshold weight**
	例：
	`switch(config)# track 1 list threshold weight`
	`switch(config-track)#`
		トラッキング対象リスト オブジェクトを設定し、トラッキング設定モードを開始します。トラッキング対象リストのステートが設定されたしきい値重みに基づいて決まることを指定します。
		track-number の範囲は 1 ～ 512 です。

Step 3	**threshold weight up** *up-value* **down** *down-value* 例： `switch(config-track)# threshold weight up 30 down 10`	トラッキング対象リストのしきい値重みを設定します。範囲は 1 ～ 255 です。

	コマンドまたはアクション	目的
Step 4	**object** *object-id* **weight** *value* 例： `switch(config-track)# object 10 weight 15`	トラッキング リストにトラッキング対象オブジェクトを追加します。*object-id* の範囲は 1 ～ 512 です。*value* の範囲は 1 ～ 255 です。デフォルトの重み値は 10 です。
Step 5	（任意） **show track** [*object-id*] 例： `switch(config-track)# show track`	オブジェクトのトラッキング情報を表示します。
Step 6	（任意） **copy running-config startup-config** 例： `switch(config-track)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、トラッキング リストのアップ重みしきい値を 30、ダウンしきい値を 10 にそれぞれ設定する例を示します。

```
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
```

この例では、オブジェクト 10 とオブジェクト 20 がアップの場合にトラッキング リストがアップになり、3 つのオブジェクトがすべてダウンの場合にトラッキング リストがダウンになります。

オブジェクト トラッキングの遅延の設定

トラッキング対象オブジェクトまたはオブジェクト トラッキング リストに対して、オブジェクトまたはリストがステートの変化を開始したときに適用する遅延を設定できます。トラッキング対象オブジェクトまたはトラッキング リストは、ステートの変化が発生したときに遅延タイマーを開始しますが、遅延タイマーが切れるまでステートの変化を認識しません。遅延タイマーが切れると、Cisco NX-OS は再びオブジェクトのステートを確認し、オブジェクトまたはリストが現在も変更されたステートのままだった場合にだけステートの変化を記録します。オブジェクト トラッキングは遅延タイマーが切れる前の中間的なステートの変化を無視します。

たとえば、インターフェイス ライン プロトコルのトラッキング対象オブジェクトがアップ ステートであり、ダウン遅延が 20 秒に設定されている場合は、ライン プロトコルがダウンになると遅延タイマーが開始します。20 秒後にライン プロトコルがダウンになっていなければ、このオブジェクトはダウン ステートになりません。

トラッキング対象オブジェクトまたはトラッキング リストには、独立したアップ遅延とダウン遅延を設定できます。遅延を削除すると、オブジェクト トラッキングからアップ遅延とダウン遅延の両方が削除されます。

遅延は任意の時点で変更できます。オブジェクトまたはリストがトリガーされたイベントから遅延タイマーをすでにカウントしている場合は、次のようにして新しい遅延が計算されます。

- 新しい設定値が古い設定値より小さい場合は、新しい値でタイマーが開始します。
- 新しい設定値が古い設定値より大きい場合は、新しい設定値から現在のタイマーのカウントダウンを引き、古い設定値を引いたものがタイマーになります。

手順の概要

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. （任意） **show track** [*object-id*]
6. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**track** *object-id* {*parameters*}
	例：
	`switch(config)# track 2 ip route 192.0.2.0/8 reachability`
	`switch(config-track)#`
		ルートのトラッキング対象オブジェクトを作成し、トラッキング コンフィギュレーション モードを開始します。*object-id* の範囲は 1 ～ 512 です。IPv4 のプレフィックス フォーマットは A.B.C.D/length です。length の範囲は 1 ～ 32 です。IPv6 のプレフィックス フォーマットは A:B::C:D/length です。length の範囲は 1 ～ 128 です。

Step 3	**track** *track-number* **list** {*parameters*}
	例：
	`switch(config)# track 1 list threshold weight`
	`switch(config-track)#`
		トラッキング対象リスト オブジェクトを設定し、トラッキング設定モードを開始します。トラッキング対象リストのステートが設定されたしきい値重みに基づいて決まることを指定します。
		track-number の範囲は 1 ～ 512 です。

	コマンドまたはアクション	目的
Step 4	**delay** {**up** *up-time* [**down** *down-time*] | **down** *down-time* [**up** *up-time*]} 例： `switch(config-track)# delay up 20 down 30`	オブジェクトの遅延タイマーを設定します。指定できる範囲は 0 ～ 180 秒です。 *track-number* の範囲は 1 ～ 512 です。
Step 5	（任意） **show track** [*object-id*] 例： `switch(config-track)# show track 3`	オブジェクトのトラッキング情報を表示します。
Step 6	（任意） **copy running-config startup-config** 例： `switch(config-track)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

例

次に、ルートのオブジェクト トラッキングを設定し、遅延タイマーを使用する例を示します。

```
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
```

次に、トラッキング リストのアップ重みしきい値を30、ダウンしきい値を10にそれぞれ設定し、遅延タイマーを使用する例を示します。

```
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
```

次に、インターフェイスがシャットダウンする前後の show track コマンドの出力に表示された遅延タイマーの例を示します。

```
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
```

非デフォルト VRF に対するオブジェクト トラッキングの設定

特定の VRF でオブジェクトをトラッキングするように Cisco NX-OS を設定できます。

始める前に

デフォルト以外の VRF が最初に作成されることを確認します。

手順の概要

1. **configure terminal**
2. **track** *object-id* {**ip** | **ipv6**} **route** *prefix/length* **reachability**
3. **vrf member** *vrf-name*
4. （任意） **show track** [*object-id*]
5. （任意） **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的

Step 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

Step 2	**track** *object-id* {**ip** | **ipv6**} **route** *prefix/length* **reachability**
	例：
	`switch(config)# track 3 ipv6 route 1::2/64 reachability`
	`switch(config-track)#`
		ルートのトラッキング対象オブジェクトを作成し、トラッキング コンフィギュレーション モードを開始します。*object-id* の範囲は 1 ～ 512 です。IPv4 のプレフィックス フォーマットは A.B.C.D/length です。length の範囲は 1 ～ 32 です。IPv6 のプレフィックス フォーマットは A:B::C:D/length です。length の範囲は 1 ～ 128 です。

Step 3	**vrf member** *vrf-name* 例： `switch(config-track)# vrf member Red`	設定されたオブジェクトのトラッキングに使用する VRF を設定します。
Step 4	（任意） **show track** [*object-id*] 例： `switch(config-track)# show track 3`	オブジェクトのトラッキング情報を表示します。
Step 5	（任意） **copy running-config startup-config** 例：	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

	コマンドまたはアクション	目的
	`switch(config-track)# copy running-config startup-config`	

例

ルートのオブジェクト トラッキングを設定し、VRF Red を使用して、そのオブジェクトの到達可能性情報を調べる例を示します。

```
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
```

次に、IPv6 ルートのオブジェクト トラッキングを設定し、VRF Red を使用して、そのオブジェクトの到達可能性情報を調べる例を示します。

```
switch# configure terminal
switch(config)# track 3 ipv6 route 1::2/64 reachability
switch(config-track)# vrf member Red
switch(config-track)# copy running-config startup-config
```

次に、トラッキング対象オブジェクト 2 を変更して、VRF Red の代わりに VRF Blue を使用してこのオブジェクトの到達可能性情報を調べるようにする例を示します。

```
switch# configure terminal
switch(config)# track 2
switch(config-track)# vrf member Blue
switch(config-track)# copy running-config startup-config
```

オブジェクト トラッキングの設定の確認

オブジェクト トラッキングの設定情報を表示するには、次のいずれかの作業を行います。

コマンド	目的
show track [*object-id*] [**brief**]	1 つまたは複数のオブジェクトについて、オブジェクト トラッキング情報を表示します。
show track [*object-id*] **interface** [**brief**]	インターフェイスベースのオブジェクト トラッキング情報を表示します。
show track [*object-id*] {**ip** | **ipv6**} **route** [**brief**]	IPv4 または IPv6 ルートベースのオブジェクト トラッキング情報を表示します。

オブジェクト トラッキングの設定例

次の例は、ルート到達可能性に対してオブジェクト トラッキングを設定し、VRF Red を使用してルートの到達可能性情報を調べる方法を示しています。

```
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
```

関連項目

オブジェクト トラッキングの関連情報については、次の項目を参照してください。

- レイヤ 3 仮想化の設定
- 『Configuring HSRP』

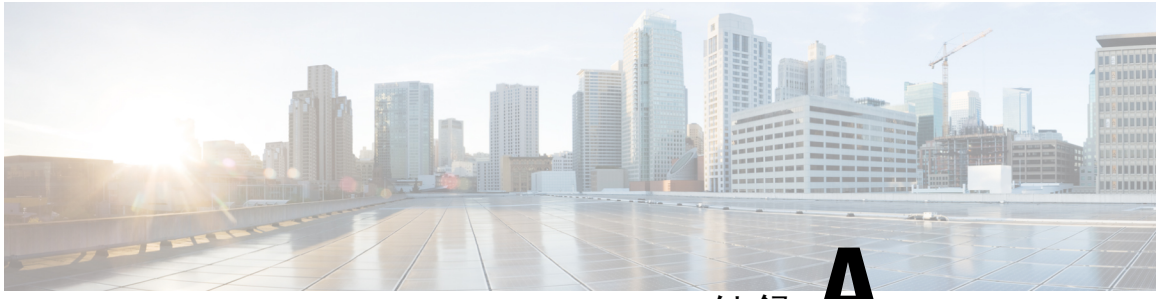
その他の参考資料

オブジェクト トラッキングの実装に関連する詳細情報については、次の項を参照してください。

- 関連資料

関連資料

関連項目	マニュアル タイトル
Embedded Event Manager の設定	『Cisco Nexus 9000 Series NX-OS System Management Configuration Guide』
IPSLA オブジェクト トラッキングの設定	『Cisco Nexus 9000 Series NX-OS IP SLAs Configuration Guide』



付録 A

Cisco NX-OS ユニキャスト機能でサポートされている IETF RFC

この付録は、Cisco NX-OS でサポートされているユニキャスト ルーティングの IETF RFC をリストにしています。

- BGP の RFC （735 ページ）
- ファーストホップ冗長プロトコルの RFC （737 ページ）
- IP サービスに関する RFC の参考資料 （737 ページ）
- IPv6 の RFC （737 ページ）
- IS-IS の RFC （738 ページ）
- OSPF の RFC （739 ページ）
- RIP の RFC （739 ページ）

BGP の RFC

RFC	タイトル
RFC 1997	*BGP* コミュニティの属性
RFC 2385	*TCP MD5* シグネチャ オプションを使用した *BGP* セッションの保護
RFC 2439	*BGP* ルート フラップ ダンピング
RFC 2519	ドメイン ルート間集約のフレームワーク
RFC 2545	*iPv6* ドメイン間ルーティングの *BGP-4* マルチプロトコル拡張の使用
RFC 2858	*BGP-4* のマルチプロトコル拡張
RFC 2918	*BGP-4* ルート更新機能

RFC	タイトル
RFC 3065	*BGP* の自律システム連合
RFC 3392	*BGP-4* による機能のアドバタイズメント
RFC 4271	ボーダー ゲートウェイ プロトコル *4 (BGP-4)*
RFC 4273	*BGP-4* の管理対象オブジェクトの定義
RFC 4456	*BGP* ルート リフレクション*:* フルメッシュ内部 *BGP (IBGP)*の代替
RFC 4486	*BGP Cease* 通知メッセージのサブコード
RFC 4724	*BGP* のグレースフル リスタート メカニズム
RFC 4760	*BGP-4* のマルチプロトコル拡張
RFC 4781	*BGP with MPLS* を使用した *BGP* のグレースフル リスタート メカニズム
RFC 4893	*4* オクテット *AS* 番号スペースの *BGP* サポート
RFC 5004	*1* つの外部から別の外部への *BGP* 最良パス移行の回避
RFC 5396[1]	自律システム *(AS)* 番号のテキスト表記
RFC 5549	*IPv6* ネクスト ホップを使用した *IPv4* ネットワーク レイヤ到達可能性情報のアドバタイズ
RFC 5668	*4-Octet AS* 指定 *BGP* 拡張コミュニティ
RFC 7606	*BGP* 更新メッセージの改訂されたエラー処理
RFC 7854	*BGP* モニタリング プロトコル（*BMP*）
draft-ietf-idr-add-paths-08.txt	*BGP* の複数パスのアドバータイズメント¥
draft-ietf-idr-bgp4-mib-15.txt	*BGP4-MIB*
draft-kato-bgp-ipv6-link-local-00.txt	*IPv6* リンクローカル アドレスを使用した *BGP4+* ピアリング
draft-ietf-idr-avoid-transition-05.txt	ベストパス遷移の回避
draft-ietf-idr-bgp4-mib-15.txt	ピア テーブル オブジェクト
draft-ietf-idr-dynamic-cap-03.txt	ダイナミック機能

[1] RFC 5396 は部分的にサポートされます。asplain と asdot 表記はサポートされますが、asdot+ 表記はサポートされません。

ファーストホップ冗長プロトコルの RFC

RFC	タイトル
RFC 2281	『*Hot Standby Redundancy Protocol*』
RFC 3768	『*Virtual Router Redundancy Protocol*』
RFC 5798	*IPv4* および *IPv6* 向け仮想ルータ冗長プロトコル（*VRRP*）バージョン *3*

IP サービスに関する RFC の参考資料

RFC	タイトル
RFC 786	*UDP*
RFC 791	*IP*
RFC 792	*ICMP*
RFC 793	*[TCP]*
RFC 826	『*ARP*』
RFC 1027	『*Proxy ARP*』
RFC 1591	『*DNS Client*』
RFC 1812	『*IPv4 routers*』
RFC 4022	*TCP-MIB*
RFC 4292	*IP-FORWARDING-TABLE-MIB*
RFC 4293	*IP-MIB*

IPv6 の RFC

RFC	タイトル
RFC 1981	IP バージョン 6 のパス MTU ディスカバリ
RFC 2374	集約可能なグローバル ユニキャスト形式
RFC 2460	インターネット プロトコル、バージョン 6（IPv6）仕様

RFC	タイトル
RFC 2464	イーサネット ネットワーク上での IPv6 パケットの送信
RFC 3021	IPv4 Point-to-Point リンクでの 31 ビット プレフィックスの使用
RFC 4191	デフォルトのルータ設定およびより固有のルート
RFC 4193	固有ローカル IPv6 ユニキャスト アドレス （注） RFC 5396 は部分的にサポートされます。セクション 3.2.2 はサポートされていません。
RFC 4291（RFC 2373 を置き換え）	IP バージョン 6 アドレス指定アーキテクチャ
RFC 4443（RFC 2463 を置き換え）	ICMPv6
RFC 4861（RFC 2461 を置き換え）	IP バージョン 6（IPv6）のネイバー探索
RFC 4862（RFC 2462 を置き換え）	IPv6 ステートレス アドレス自動設定
RFC 6106	DNS 設定の IPv6 ルータ アドバタイズメント オプション
RFC 2526	予約済みの IPv6 サブネットエニーキャストアドレス

IS-IS の RFC

RFC	タイトル
RFC 1142	『*OSI 10589 Intermediate system to intermediate system intra-domain routing exchange protocol*』
RFC 1195	『*Use of OSI IS-IS for routng in TCP/IP and dual environment*』
RFC 2763、RFC 5301	『*Dynamic Hostname Exchange Mechanism for IS-IS*』
RFC 2966、RFC 5302	『*Domain-wide Prefix Distribution with Two-Level IS-IS*』
RFC 2972	『*IS-IS Mesh Groups*』
RFC 3277	『*IS-IS Transient Blackhole Avoidance*』
RFC 3373、RFC 5303	『*Three-Way Handshake for IS-IS Point-to-Point Adjacencies*』

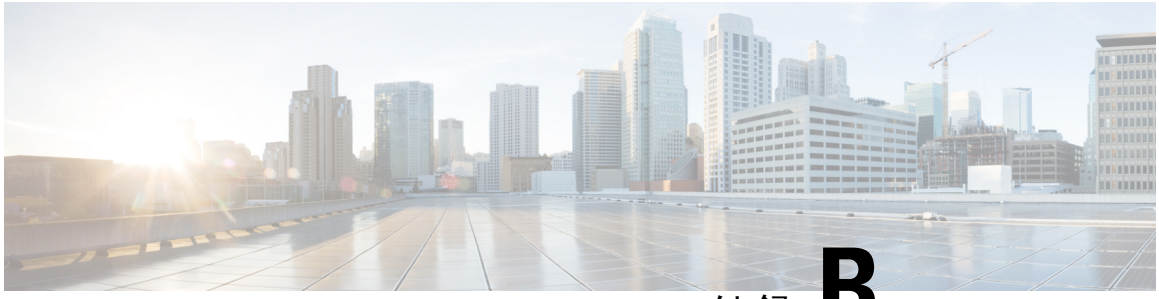
RFC	タイトル
RFC 3567、RFC 5304	『*IS-IS Cryptographic Authentication*』
RFC 3784、RFC 5305	『*IS-IS Extensions for Traffic Engineering*』
RFC 3847、RFC 5306	『*Restart Signaling for IS-IS*』
RFC 4205、RFC 5307	『*IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching*』
draft-ietf-isis-igp-p2p-over-lan-06.txt	『*Internet Draft Point-to-point operation over LAN in link-state routing protocols*』

OSPF の RFC

RFC	タイトル
RFC 2328	『*OSPF Version 2*』
RFC 2370	『*The OSPF Opaque LSA Option*』
RFC 2740	『*OSPF for IPv6*』
RFC 3101	『*The OSPF Not-So-Stubby Area (NSSA) Option*』
RFC 3137	『*OSPF Stub Router Advertisement*』
RFC 3623	『*Graceful OSPF Restart*』
RFC 5709	『*OSPFv2 HMAC-SHA Cryptographic Authentication*』
draft-ietf-ospf-ospfv3-graceful-restart-04.txt	『*OSPFv3 Graceful Restart*』

RIP の RFC

RFC	タイトル
RFC 2082	『*RIP-2 MD5 Authentication*』
RFC 2453	『*RIP Version 2*』



付録 B

Cisco NX-OS レイヤ 3 ユニキャスト機能の設定の制限

• Cisco NX-OS レイヤ 3 ユニキャスト機能の設定の制限 （741 ページ）

Cisco NX-OS レイヤ 3 ユニキャスト機能の設定の制限

設定制限は『Cisco Nexus 9000 シリーズ NX-OS 検証済みスケーラビリティ ガイド』にまとめられています。

Cisco NX-OS レイヤ 3 ユニキャスト機能の設定の制限

INDEX

{ip | ipv6} address 311–312
{ip | ipv6} bandwidth eigrp 270–271
{ip | ipv6} bandwidth-percent eigrp 270, 272
{ip | ipv6} delay eigrp 270, 272
{ip | ipv6} distribute-list eigrp 270, 272
{ip | ipv6} hello-interval eigrp 268
{ip | ipv6} hold-time eigrp 268
{ip | ipv6} next-hop-self eigrp 270, 272
{ip | ipv6} passive-interface eigrp 270, 272
{ip | ipv6} prefix-list 600–601
{ip | ipv6} router eigrp 273–274
{ip | ipv6} router isis 292, 311–312
{ip | ipv6} split-horizon eigrp 268–269
{ip | ipv6} 600–601
アドミニストレーティブ ディスタンス 526
address 692–693, 704–705, 707, 710–711
object 724–728
ガイドライン 512
　RIPng 512
確認 522
　RIPng 522
仮想化 519
　構成 519
仮想化のサポート 511
　RIPng 用 511
関連資料 57
　IPv4 57
キー 131, 133
消去 584
グレースフル リスタート 309
高可用性 511
　RIPng 511
作成 513
　RIPng インスタンス 513
しきい値重み 727
シャットダウン 253, 291, 338–340, 694–701, 704, 706–708
重量 339–340
スタティック ルート 10
stub 257–258
スタブ ルーティング 8
スプリット ホライズン 510
制限事項 512
　RIPng 512
設定例 523
　RIPng 523
説明 339–340, 380–381, 446, 704–705
ダイナミック ルーティング プロトコル 10
timers 339–340, 380–381, 385–387, 674
調整 521
　RIPng 521
テスト転送 580–581
デフォルト設定 512
　RIPng 512
転送の消去 581
等コストマルチパス（ECMP） 510
トラックなし 721–722
名前 674
認証 145–146
認証テキスト 696
net 289, 311–312
network 336–337
ハードウェア IP 収集スロットルの最大タイムアウト 54
パケット交換 4
パスワード 380–381
プライオリティ 671–673, 694–695, 704–705
リンクステート プロトコル 11
ルート再配布 489
ルート集約 489
ルートのフィルタリング 489, 510
ロード バランシング 510

A

additional-paths receive 401–402
additional-paths selection route-map 404
additional-paths send 401–402
address-family {ipv4 | ipv6} {unicast | multicast 451
address-family {ipv4|ipv6} {unicast|multicast} 336–337, 339–340
address-family {ipv4 | ipv6} {unicast | multicast} 411–413, 423–424, 426–427
address-family {ipv4 | ipv6} unicast 264–266, 300–302, 439–440
address-family {ipv4 | ipv6} {multicast | unicast} 383–386, 390
address-family 255–256, 260, 415
address-family ipv4 unicast 307–308, 492–493, 499, 503–504

address-family ipv6 unicast 193–195, 197–198, 203–204, 206–210, 212–213, 307–308, 513–514, 519–520
adjacency-check 313–314
timers advertise 704, 706
advertise-active-only 383
advertise-map 424
advertisement interval 697–698
aggregate-address 374
allowas in 447
area 131–132, 137–142, 145, 151–152, 193–198, 201–202, 207–208
as-override 448
authentication key-chain 255–256, 294–295
authentication mode md5 255–256
authentication-key 145–146
autonomous-system 250–251, 265

B

bgp confederation peers 411

C

capability additional-paths receive 400–401
capability additional-paths send 400–401
clear bgp {ipv4 | ipv6} {unicast | multicast} flap-statistics [vrf 347
clear bgp {ipv4 | ipv6} {ユニキャスト | マルチキャスト 394–395
clear bgp all dampening 347
clear bgp all flap-statistics 347
clear bgp all 347
clear bgp dampening 349
clear bgp flap-statistics 350
clear bgp 347
clear ip eigrp redistribution 263
clear ip mbgp dampening 351
clear ip mbgp flap-statistics 351
clear ip mbgp 350
clear ip rip statistics 507
clear isis 289–290
clear rip policy statistics redistribute 507
clear routing 584
clear vrrpv3 statistics 712
ipv6 rip 統計をクリア 523
client-to-client reflection 412
cluster-id 411–412
confederation identifier 411
continue 612–613

D

dampening 417
デッド間隔 145–146, 201–202
default-information originate 148, 203–204, 269–270, 302, 439–440, 499–500
default-metric 148–149, 203–204, 260–261, 426–427, 499–500
default-originate 447
delay 729–730
delay restore 185
disable-connected-check 341, 405
disable-peer-as-check 448
ポリシーバッチ処理の無効化 400
distance 127, 188–189, 269–270, 289–290, 446, 492–493, 513–514
distribute {level-1 | level-2} into {level-1 | level-2} 302–303
distribute-list 276
dont-capability-negotiate 399
dscp 420

E

ebgp-multihop 341, 408
enforce-first-as 444
enhanced-error 442

F

feature bgp 335
feature eigrp 249
feature hsrp 662
feature interface vlan 529–530
feature isis 288
feature ospf 124
feature ospfv3 186–187
feature pbr 631–634, 643–644
feature rip 491–492, 513
feature vrrp 691–692
feature vrrpv3 703
filter-list 448
flush-routes 252–253

G

gateway protocols 10
graceful-restart 161, 215–216, 266–267, 478–479
graceful-restart grace-period 161, 215–216
graceful-restart helper-disable 161, 215–216
graceful-restart planned-only 161, 215–216
graceful-restart t3 manual 309–310
graceful-restart-helper 478–479

H

hardware ip glean throttle maximum 53
hardware ip glean throttle 52
hello-interval 145–146, 201–202
hostname dynamic 298
hsrp timers extended-hold 677
hsrp version {1 | 2} 663
hsrp 663–664, 666, 669, 671–672
hsrp version 2 666

I

inherit peer 385–386, 390
inherit peer-policy 383–386
inherit peer-session 380–381, 385–386
interface ethernet 31–33
インターフェイス 255–256, 515
　RIPng の 構成 515
interface-vlan 529–530
ip 247, 663–664, 666
ip | ipv6} offset-list eigrp 270, 272
ip arp address 43
ip arp gratuitous {request | update} 48
ip as-path access-list 604
ip authentication key-chain eigrp 255–256
ip authentication mode eigrp 255, 257
ip autoconfig 666–667
ip community-list expanded 609
ip community-list standard 609
ip directed-broadcast 51–52
ip domain-list 108, 110–111, 547–548
ip domain-lookup 108–109
ip extcommunity-list expanded 610–611
ip extcommunity-list standard 610–611
ip ospf authentication key-chain 133–134
ip ospf authentication 133–134
ip ospf authentication-key 131–134
ip ospf cost 129–130
ip ospf dead-interval 129–130, 158–159
ip ospf hello-interval 129–130, 158–159
ip ospf message-digest-key 131–134
ip ospf mtu-ignore 129–130
ip ospf passive-interface 129–130
ip ospf retransmit-interval 158–159
ip passive-interface eigrp 254
ip proxy arp 43–44
ip rip authentication keychain 496
ip rip authentication mode 496
ip rip metric-offset 506
ip rip passive-interface 497
ip rip poison-reverse 498
ip rip route-filter 506
ip rip summary-address 498
router eigrp 250, 255–256, 260, 262–266, 273–274
ip router eigrp 250–251, 255–256
ip router ospf 129, 163–164, 545, 547
ip router rip 494–495, 503–504
ip source 54–55
ip summary-address eigrp 259
ip tcp path-mtu-discovery 50–51
ip address 31–33, 129, 163–164, 503–504, 530, 544–545, 547, 707, 710
ip domain-name 108, 110
ip host 108
ip name-server 108–111
ip route 439–440, 528–532, 543
IPv4 57
　関連資料 57
ipv6 247
ipv6 address use-link-local-only 83
ipv6 authentication key-chain eigrp 255–256
ipv6 authentication mode eigrp 255, 257
ipv6 ospfv3 217, 219
ipv6 passive-interface eigrp 254
show ipv6 rip 515–516, 519–520, 523
ipv6 route 528–529, 531–532
ipv6 router eigrp 250–251, 255–256
ipv6 router ospfv3 190–191, 200
ipv6 summary-address eigrp 259
ipv6 アドレス 82–83, 190–191, 217–218, 519–520, 666
ipv6 rip poison-reverse 517
ipv6 rip ルートフィルタ 522
ipv6 ルータ rip 515–516, 519–520
is-type {level-1 | level-2 | level-1-2} 289
isis authentication key-chain 296
authentication-check {level-1 | level-2} 294–295
isis authentication-check {level-1 | level-2} 296
authentication-type {cleartext | md5} {level-1 | level-2} 294–295
isis authentication-type {cleartext | md5} {level-1 | level-2} 296
isis circuit-type {level-1 | level-2 | level-1-2} 292–293
isis csnp-interval 313–314
isis hello-interval 313–314
isis hello-multiplier 313, 315
isis hello-padding 300
isis lsp-interval 313, 315
isis mesh-group 297
isis metric 292–293
isis passive {level-1 | level-2 | level-1-2} 292–293
isis priority 297–298
isis shutdown 294

L

local-as 410
log-adjacency-changes 127, 188–189, 250–251, 289–290
log-neighbor-changes 445–446
log-neighbor-warnings 250–251
low-memory exempt 446
lsp-gen-interval 313
lsp-mtu 289–290

M

mac address 710–711
mac-address 668
match ip address prefix-list 155–156
match ip route-source 184–185
match ip route-source prefix-list 155–156, 184, 209, 211
match ipv6 address prefix-list 184, 209, 211
match ipv6 route-source 184
match ipv6 address 185

match route-type 155–156, 184–185, 209, 211
match-address 704–705
max-lsp-lifetime 313–314
max-metric router-lsa 153–154
maxas-limit 409
maximum routes 582
maximum-paths 127–128, 163–164, 188–189, 217–218, 264–265, 289–290, 417–418, 492–493, 514, 545–546
maximum-peers 388
maximum-prefix 383
medium {broadcast | p2p} 292
message-digest-key 145–146
metric direct 0 501, 517–518
metric max-hops 270
metric weights 270
metric-style transition 313–314
metrics rib-scale 269
metric version 64bit 269

N

neighbor 342, 380–381, 383–386, 389, 412–415, 423–424, 451, 481
neighbor-down fib-accelerate 399
next-hop-self 395–396, 414
next-hop-third-party 395–396
nexthop route-map 397
nexthop suppress-default-resolution 397
no {ip | ipv6} route 529
no adjacency-check 307–308
no adjacency-checkg 307
no fast-external-fallover 409
no preempt 699
no shutdown 663–664, 666–667, 692–701
nsf await-redist-proto-convergence 270–271

O

ospfv3 cost 190–191
ospfv3 dead-interval 190–191
ospfv3 hello-interval 190–191
ospfv3 インスタンス 190, 192
ospfv3 mtu-ignore 190, 192
ospfv3 network 190, 192
ospfv3 passive-interface 190, 192
ospfv3 priority 190, 192
ospfv3 retransmit-interval 213–214
ip ospf transmit-delay 158, 160
ospfv3 transmit-delay 213–214

P

passive-interface default 127–128, 188–189
path-attribute discard 441
path-attribute treat-as-withdraw 441
preempt 671–672, 674, 704–705
prefix-list 448

R

redistribute 148, 203–204, 260–263, 302, 304, 499, 503–504
redistribute {direct | {eigrp | isis | ospf | ospfv3 | rip} 426–427
redistribute bgp 206
redistribute static route-map allow 439–440
redistribute maximum-prefix 206, 262–263, 304
reference-bandwidth 289–290
reload module 119, 182, 285
reload 34–38, 85–89, 346
remove-private-as 374, 447
restart bgp 338
restart eigrp 253
restart ospf 119, 162
restart ospfv3 182, 217
restart rip 494, 515
restart isis 285, 291
retransmit-interval 145–146, 201–202
RIPng 509, 511–512, 521–523
　ガイドライン 512
　確認 522
　仮想化のサポート 511
　高可用性 511
　制限事項 512
　設定例 523
　説明 509
　調整 521
　デフォルト設定 512
　有効化 512
RIPng インスタンス 513, 515
　再起動 515
　作成 513
RIPng 統計 523
　表示 523
RIPng の 構成 515
　インターフェイス上 515
route-map 155, 209–210, 415–416, 606–607, 612
route-map allow permit 439
route-reflector-client 412–413, 415
router bgp 336, 339, 342, 380, 382–383, 385, 389, 411–412, 414–415, 423–424, 426, 439–440, 451, 478, 481
router isis 289, 294–295, 300–302, 304, 307–309, 311
router ospf 131–132, 137–138, 141, 145, 148, 151–155, 158, 161, 163–164, 545–546
router ospfv3 188–189, 193–195, 197–198, 201, 203–204, 206–210, 212–213, 215, 217–218
router rip 492–493, 499, 501, 503–504, 513–514, 517–520
router-id 188–189, 336–337, 445
routing-context vrf default 549
routing-context vrf 549

S

send-community 448
send-community extended 448
set distance 155–156, 209, 211
set ip next-hop peer-address 414
set ipv6 next-hop peer-address 414
set next-hop 414
set-attached-bit 299
set-overload-bit {always | on-startup 299
show 536, 549, 579
show {ip | ipv6} adjacency 585
show {ip | ipv6} eigrp route-map statistics redistribute 260–261
show {ip | ipv6} eigrp 275
show {ip | ipv6} route 579, 585
show {ip | ipv6} routing 579
show {ip | ipv6} static-route 528–529, 533
show {ip | ipv6} static-route track-table 533
show {ip | ipv6} 250, 252, 531–532, 601, 604
show {ipv | ipv6} bgp 353
show {ipv | ipv6} mbgp 353
show {ipv4 | ipv6} bgp 484
show {ipv4 | ipv6} mbgp 484
show bgp {ipv4 | ipv6 | vpnv4 | vpnv6} {unicast | multicast} 483
show bgp {ipv4 | ipv6} {unicast | multicast} 351–352, 354, 415–416, 482–483, 485
show bgp {ipv4 | ipv6} unicast injected-routes 485
show bgp {ipv4 | ipv6} unicast path-attribute discard 442
show bgp {ipv4 | ipv6} unicast path-attribute unknown 442
show bgp ipv6 unicast 402–404, 442
show bgp {ipv4 | ipv6} {unicast | multicast} neighbors 339, 341–342, 412–413
show bgp {ipv4|ipv6} unicast neighbors 390, 484
show bgp all 336–337, 351, 482
show bgp convergence 351, 482
show bgp ipv4 multicast neighbors 424–425
show bgp ipv4 unicast neighbors 388, 424–425
show bgp ipv6 unicast neighbors 424–425
show bgp neighbor 381–382, 384, 387, 400–402
show bgp peer-policy 353, 383–384, 483
show bgp peer-session 353, 380–381, 483
show bgp peer-template 353, 385, 387, 484
show bgp process 353, 484
show bgp sessions 354, 485
show bgp statistics 354, 485
show bgp vrf 351
show consistency-checker 580–581
show feature 124, 187, 249, 288, 335–336, 491–492, 513, 631–632
show fhrp 704, 706–708, 711
show forwarding {ip | ipv4 | ipv6} route 585
show forwarding {ipv4 | ipv6} adjacency module 561
show forwarding {ipv4 | ipv6} route module 561
show forwarding adjacency 584
show forwarding distribution {clients | fib-state} 584
show forwarding interfaces module 585
show forwarding route summary 34–38, 84–89
show forwarding 580–581
show hosts 108–112
show hsrp 664, 666–667, 669–670, 677–678
show hsrp delay interface 678
show hsrp group 678
show hsrp interface 671–672, 678
show interface 711
show ip rip 492–495, 503, 505, 507, 513–514
show ip adjacency summary 56
show ip adjacency 56
show ip arp statistics 56
show ip arp summary 56
show ip arp 56
show ip bgp neighbors 390, 484
show ip community list 609
show ip community-list 610–611, 621
show ip eigrp neighbor detail 257–258
show ip ext community-list 621
show ip interface 32–33, 56
show ip load-sharing 562, 564
show ip ospf interface 131
show ip ospf neighbor 131
show ip ospf policy statistics area 137, 166
show ip ospf statistics 166
show ip ospf summary-address 151–152
show ip ospf virtual-link 145
show ip policy statistics redistribute 166
show ip rip instance 507
show ip rip route 499–500
show ip route 530
show ip ospf traffic 166
show ipv6 adjacency 103
show ipv6 interface 82–83, 103
show ipv6 ospfv3 188–191, 200, 215–216
show ipv6 ospfv3 memory 234
show ipv6 ospfv3 policy statistics area 193–194, 234
show ipv6 ospfv3 policy statistics redistribute 234
show ipv6 ospfv3 statistics 234
show ipv6 ospfv3 summary-address 207–208
show ipv6 ospfv3 traffic 234
show ipv6 ospfv3 virtual-link 201–202
show ipv6 routers interface 390, 484
show ip static-route vrf 533
show isis 289–290, 292–293, 300–303, 311–312, 315–316
show ip ospf 129–133, 135, 138–139, 141–142, 158, 160–162
show platform fib 14
show platform forwarding 14
show policy 645
show prefix-list 621
show route-map brief 622
show route-map 621, 646
show routing hash 562, 564
show routing 583, 585
show running-config bgp 478, 480
show running-config isis 304–305, 307–310
show running-config ospfv3 206–207

show running-config rip 501–502, 517–518
show running-config eigrp 262–263
show running-configuration bgp 353, 484
show running-configuration eigrp 275
show running-configuration isis 316
show running-configuration rip 507, 523
show tech-support isis 316
show track 720–732
show vrf 543–545, 550
show vrrp statistics 712
show vrrpv3 statistics 712
show vrrpv3 704, 706–708
show bgp 482
show bgp paths 353
ipv6 rip インスタンスを表示します 522
show vrrp 692–701, 711
snmp-server host 547–548
soft-reconfiguration inbound 394–395
spf-interval [level-1 | level-2 313–314
split horizon with poison reverse 516
　構成 516
summary-address 151–152, 207–208, 300–301
suppress-fib-pending 374, 423
suppress-inactive 448
system pic enable 346
system pic-core 346
system routing max-mode host 34, 84–85
system routing max-mode l3 38, 88–89
system routing mode hierarchical 64b-alpm 36–37, 87
system routing non-hierarchical-routing 35, 86
system switchover 119, 182, 285

T

table-map 155, 209–210
template peer 385–386
template peer-session 380, 383
threshold percentage up 725–726
timers [bestpath-delay 445
timers active-time 270–271
timers basic 506, 522
timers lsa-arrival 158, 212–213
timers lsa-group-pacing 158, 212–213
timers nsf converge 266–267
timers nsf route-hold 266–267
timers nsf signal 266–267
timers prefix-peer-timeout 388, 478–479
timers throttle lsa 158–159, 212–213
timers throttle spf 158–159
track interface 700–701
track 671–672, 720–727, 729, 731
transmit-delay 145–146, 201–202
transport connection-mode passive 446

U

update-source 414–415, 447

V

vrf 163–164, 217–218, 311, 481, 503–504, 519–520, 545–546
vrf context 110, 163, 217–218, 273, 310–311, 481, 503, 519, 531–532, 543, 547–548, 582
vrf member 163–164, 217–218, 273–274, 311–312, 503–504, 519–520, 544–546, 731
vrrp 692–701
vrrp2 704, 706
vrrpv3 704–705, 707
vrrs leader 704, 706
show vrrs pathway 710–711
vrrs pathway 710

W

write erase boot 541
write erase 541

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。