ルーティング

この章では、拡張または拡張サービスの設定について説明します。製品アドミニストレーションガイドには、システム上での基本サービスの設定例と手順が示されています。以下で説明する手順を使用する前に、サービスモデルに最適な設定例を選択し、そのモデルに必要な要素を設定する必要があります。

この章は、次の項で構成されています。

ルーティング ポリシー

この項では、ルーティングポリシーを定義するために必要な要素を設定する方法について説明します。ルーティングポリシーは、特定のネットワーク展開要件を満たすために、システムとの間でルートを変更およびリダイレクトします。

ルーティングポリシーを設定するには、次の構成要素を使用します。

- **ルートアクセスリスト**：ルーティングポリシーの基本的な構成要素です。ルートアクセスリストは、IP アドレスの範囲に基づいてルートをフィルタ処理します。
- **IP プレフィックスリスト**：より高度なルーティングポリシーの要素です。IP プレフィックスリストは、IP プレフィックスに基づいてルートをフィルタ処理します。
- **AS パスアクセスリスト**：ボーダー ゲートウェイ プロトコル（BGP）のルーティングに使用される基本的な構成要素です。これらのリストは、自律システム（AS）のパスをフィルタ処理します。

- **ルートマップ**：ルートマップは、ルーティングプロトコルによるルートの選択またはルートのアドバタイズメントの際、およびルーティングプロトコル間のルートの再配布において、ルートマップの詳細な制御を提供します。このレベルの制御では、IPプレフィックスリスト、ルートアクセスリスト、およびASパスアクセスリストを使用して、IPアドレス、アドレスの範囲、および自律システムパスを指定します。

IP プレフィックスリストの作成

IP プレフィックスリストを作成するには、次の設定例を使用します。

```
config
   context context_name
      ip prefix-list name list_name { deny | permit } network_address/net_mask
```

注：

- IP プレフィックスリストを [deny]、[permit]、または [match any prefix] に設定します。
- IPv4 ドット付き 10 進数および IPv6 コロン区切り 16 進数のアドレスがサポートされています。
- 「設定の確認と保存」の章の説明に従って、設定を保存します。

ルートアクセスリストの作成

ルートアクセスリストを作成するには、次の手順を実行します。

```
config
   context context_name
      route-access-list { extended identifier } { deny | permit } [ ip
address ip_address ]
      route-access-list named list_name { deny | permit } { ip_address/mask |
 any } [ exact-match ]
route-access-list
standard identifier { permit | deny ) { ip_address
wildcard_mask | any |network_address }
```

注：

- コンテキストごとに最大 64 のアクセスリストがサポートされます。
- 各ルートアクセスリストに対して最大 16 のエントリを定義できます。
- 「設定の確認と保存」の章の説明に従って、設定を保存します。

AS パスアクセスリストの作成

AS パスアクセスリストを作成するには、次の手順を実行します。

```
config
   context context_name
      ip as-path access-list list_name [ { deny | permit } reg_expr ]
```

注：

- 「設定の確認と保存」の章の説明に従って、設定を保存します。

ルートマップの作成

ルートマップを作成するには、次の設定例を使用します。

```
config
   context context_name
      route-map map_name { deny | permit } seq_number
```

注：

- ルートマップを設定するには、ルート マップ コンフィギュレーション モードで **match** コマンドと **set** コマンドを使用します。これらのコマンドの詳細については、『*Command Line Interface Reference*』を参照してください。
- 「設定の確認と保存」の章の説明に従って、設定を保存します。

設定例

次に、2つのルートアクセスリストを作成し、それらをルートマップに適用し、そのルートマップを BGP ルータネイバーに使用する設定の例を示します。

```
config
   context isp1
      route-access-list named RACLin1a permit 88.151.1.0/30
      route-access-list named RACLin1b permit 88.151.1.4/30
      route-access-list named RACLany permit any
      route-map RMnet1 deny 100
         match ip address route-access-list RACLin1a
         #exit
         route-map RMnet1 deny 200
         match ip address route-access-list RACLin1b
         #exit
      route-map RMnet1 permit 1000
         match ip address route-access-list RACLany
         #exit
      router bgp 1
         neighbor 152.20.1.99 as-path 101
         neighbor 152.20.1.99 route-map RMnet1 in
```

スタティック ルーティング

システムでは、コンテキストごとにスタティック ネットワークルートの設定がサポートされています。次のように指定して、ネットワークルートを定義します。

- ルートの IP アドレスとマスク
- ルートが使用する必要がある現在のコンテキスト内のインターフェイスの名前
- ネクスト ホップの IP アドレス

重要 VPC-DI では、IPv6 プレフィックス長が /12 未満で、/64 と/128 の範囲のスタティックルートはサポートされていません。

コンテキストへのスタティックルートの追加

コンテキストコンフィギュレーションにスタティックルートを追加するには、現在のコンテキストで設定されているインターフェイスの名前を把握している必要があります。現在のコンテキスト（Exec モード）のインターフェイスを一覧表示するには、**show ip interface** コマンドを使用します。

次の例に示すように、現在のコンテキストで設定されているすべてのインターフェイスの情報が表示されます。

```
[local]host_name# show ip interface
Intf Name: Egress 1
Description:
IP State: Up (Bound to slot/port untagged ifIndex 402718721)
IP Address: 192.168.231.5
Subnet Mask: 255.255.255.0
Bcast Address: 192.168.231.255
MTU: 1500
Resoln Type: ARP          ARP timeout: 3600 secs
L3 monitor LC-port switchover: Disabled
Number of Secondary Addresses: 0
Total interface count: 1
```

出力例に示されているように、各インターフェイスの最初の情報行には、現在のコンテキストのインターフェイス名が一覧表示されます。この例では、*Egress 1* という名前のインターフェイスが 1 つあります。

```
config
   context context_name
      ip route { ip_address [ ip_mask ] | ip_addr_mask_combo } { next-hop
next_hop_address | egress_name [ precedence precedence [ cost cost ]
```

注：

- コンテキストごとに最大 1,200 のスタティックルートを設定できます。「設定の確認と保存」の章の説明に従って、設定を保存します。

コンテキストからのスタティックルートの削除

コンテキストの設定からスタティックルートを削除するには、次の設定例を使用します。

```
config
   context context_name
       no ip route { ip_address ip_mask | ip_addr_mask_combo } next_hop_address
egress_name [ precedence precedence ] [ cost cost ]
```

注記

- 「設定の確認と保存」の章の説明に従って、設定を保存します。

OSPF ルーティング

この項では、システムでの Open Shortest Path First （OSPF） ルーティングの概要とその実装を示します。また、基本的な OSPF 機能を有効にする方法と、より複雑な設定に使用できるコマンドのリストを示します。

この機能を使用するには、ライセンスキーを購入してインストールする必要があります。ライセンスの詳細については、シスコのアカウント担当者にお問い合わせください。

重要 リカバリ中にダイナミック ルーティング プロトコルから転送エントリが削除された場合は、動的に学習された転送エントリがシステムタスクのリカバリ時にシステム転送テーブルに誤って残る可能性があります。

重要 VPC-DIでは、IPv6 プレフィックス長が /12 未満で、/64 ～ /128 の範囲の OSPF ルートはサポートされていません。

OSPF バージョン 2 の概要

OSPF は、内部ゲートウェイプロトコル（IGP）を使用し、IP パケットヘッダー内の接続先 IP アドレスのみに基づく最短パスを最初に使用して IP パケットをルーティングするリンクステート ルーティング プロトコルです。OSPF ルーテッド IP パケットは、ネットワークを通過するときに追加のプロトコルヘッダー内にカプセル化されません。

自律システム（AS）またはドメインは、共通のルーティング インフラストラクチャ内のネットワークのグループとして定義されます。

OSPF は、AS 内のトポロジの変化（ルータインターフェイスの障害など）を迅速に検出し、コンバージェンスの期間後に新しいループフリールートを計算するダイナミック ルーティング プロトコルです。このコンバージェンスの期間は短く、関与するルーティングトラフィックは最小限となります。

リンクステート ルーティング プロトコルでは、各ルータは、自律システムのトポロジを記述する、リンクステート データベースと呼ばれるデータベースを維持します。参加している各ルータには同一のデータベースが備わっています。このデータベースの各エントリは、特定の

ルータのローカル状態（ルータの使用可能なインターフェイスや到達可能なネイバーなど）です。ルータは AS 全体にローカル状態であることをフラッディングにより配信します。

すべてのルータが同じアルゴリズムを並行して実行します。各ルータはそれ自体を AS 内の各接続先へのルートとした最短パスのツリーをリンクステートデータベースから構築します。外部から取得したルーティング情報は、リーフとしてツリーに表示されます。ルートのコストは、1 つの無次元メトリックで記述されます。

OSPF では、一連のネットワークをまとめてグループ化できます。このようなグループ化をエリアと呼びます。このエリアのトポロジは AS の残りの部分からは隠されているため、ルーティングトラフィックの大幅な削減が可能になります。また、エリア内のルーティングはエリア自体のトポロジによってのみ決定されるため、不正なルーティングデータからエリアを保護できます。エリアとは、IP サブネット化ネットワークの汎化です。

OSPF では IP サブネットを柔軟に設定でき、OSPF によって配信された各ルートに接続先とマスクを持たせることができます。同じ IP ネットワーク番号の 2 つの異なるサブネットでは、サイズが異なる（つまり、マスクが異なる）場合があります。これは一般に可変長サブネットと呼ばれています。パケットは最適な（最長または最も具体的な）一致にルーティングされます。ホストルートは、マスクが「すべて 1」（0xffffffff）であるサブネットと見なされます。

OSPF トラフィックは、認証されるかまたは非認証となります。また、認証、簡易/クリアテキストのパスワード、または MD5 ベースのパスワードは使用できません。つまり、信頼できるルータのみが AS ルーティングに参加できることを意味します。さまざまな認証スキームを指定でき、実際には、IP サブネットごとに個別の認証スキームを設定できます。

外部から取得したルーティングデータ（BGP などの外部プロトコルから学習したルートなど）は AS 全体にアドバタイズされます。この外部から取得したデータは、OSPF のリンクステートデータとは別に保持されます。

また、各外部ルートはアドバタイズルータによってタグ付けされ、AS の境界にあるルータ間で追加情報を渡すこともできます。

OSPF は、リンクステート アルゴリズムを使用して、すべての既知の宛先までの最短パスを構築および計算します。

OSPFv2 設定の基本

この項では、基本的な OSPF ルーティングの実装方法を説明します。

特定のコンテキストの OSPF ルーティングの有効化

特定のコンテキストに対して OSPF ルーティングを有効にするには、次の設定例を使用します。

```
config
  context context_name
    router ospf
    end
```

注：

- 「設定の確認と保存」の章の説明に従って、設定を保存します。

特定のインターフェイスでの OSPF の有効化

OSPF を有効にした後、実行するネットワークを指定します。次のコマンドを使用して、OSPF を有効にします。

network *network_ip_address*/*network_mask* **area** { *area_id* | *area_ip_address* }

重要 システム上の OSPF のデフォルトコストは 10 です。コストを変更するには、『*Command Line Interface Reference*』の「*Ethernet Interface Configuration Mode Commands*」の章に記載されている **ip ospf cost** コマンドを参照してください。

注：

- 「設定の確認と保存」の章の説明に従って、設定を保存します。

OSPF へのルートの再配布（オプション）

ルートを OSPF に再配布するということは、ルートタイプ、メトリック、ルートマップ内のルールなど、指定された基準を満たす別のプロトコルからのすべてのルートが、OSPFv2 プロトコルを使用してすべての OSPF エリアに再配布されることを意味します。この設定は、オプションです。

```
config
   context context_name
      router ospf
         redistribute { connected   | static }
         end
```

注：

- 「設定の確認と保存」の章の説明に従って、設定を保存します。

OSPF 設定パラメータの確認

OSPF ルータの設定を確認するには、次のコマンドを使用し、画面出力で **router OSPF** というセクションを探します。

show config context *ctxt_name* **[verbose]**

OSPFv3 ルーティング

この項では、システムでの Open Shortest Path First バージョン 3（OSPFv3）のルーティングとその実装の概要を示します。また、基本的な OSPFv3 機能を有効にする方法と、より複雑な設定に使用できるコマンドのリストを示します。

重要　VPC-DI では、IPv6 プレフィックス長が /12 未満で、/64 ～ /128 の範囲の OSPFv3 ルートはサポートされていません。

OSPFv3 の概要

OSPF バージョン 3 の大部分は OSPF バージョン 2 と同じです。OSPFv3 は IPv6 ルーティングプレフィックスと大型の IPv6 アドレスをサポートするように OSPF バージョン 2 が拡張されています。OSPFv3 は、OSPFv3 ルーティングドメイン内の IPv6 ルートを動的に学習し、アドバタイズ（再配布）します。

OSPFv3 では、ルーティング プロセスを明示的に作成する必要はありません。インターフェイス上で OSPFv3 を有効にすると、ルーティングプロセスとそれに関連する設定が作成されます。

OSPFv3 設定の基本

この項では、基本的な OSPF ルーティングの実装方法を説明します。

特定のコンテキストにおける OSPFv3 ルーティングの有効化

特定のコンテキストに対して OSPF ルーティングを有効にするには、次の設定例を使用します。

```
config
  context context_name
    router ospfv3
    end
```

注：

- 「設定の確認と保存」の章の説明に従って、設定を保存します。

特定のインターフェイスでの OSPFv6 の有効化

OSPFv3 を有効にした後は、実行するエリアを指定します。次のコマンドを使用して、OSPFv3 を有効にします。

```
area { area_id | area_ip_address } [ default-cost dflt-cost ] [ stub stub-area ] [ virtual-link vl-neighbor-ipv4address ]
```

重要　システム上の OSPFv3 のデフォルトコストは 10 です。コストを変更するには、『*Command Line Interface Reference*』の「*Ethernet Interface Configuration Mode Commands*」の章に記載されている **ipv6 ospf cost** コマンドを参照してください。

注：

- 「設定の確認と保存」の章の説明に従って、設定を保存します。

OSPFv3 へのルートの再配布（オプション）

ルートを OSPFv3 に再配布するということは、ルートタイプ、メトリック、ルートマップ内のルールなど、指定された基準を満たす別のプロトコルからのすべてのルートが、OSPFv3 プロトコルを使用してすべての OSPF エリアに再配布されることを意味します。この設定は、オプションです。

```
config
  context context_name
    router ospf3
      redistribute { connected | static }
      end
```

注：

- 「設定の確認と保存」の章の説明に従って、設定を保存します。

OSPFv3 設定パラメータの確認

OSPF ルータの設定を確認するには、次のコマンドを使用して、画面出力で「**router ipv6 ospf**」というラベルの付いたセクションを探します。

```
show config context ctxt_name [ verbose ]
```

等コストマルチパス（ECMP）

システムは、ルーティングプロトコルの ECMP をサポートしています。ECMP は、1 つのルートの負担を軽減するために、コストが同じである複数のルートにトラフィックを分散します。

ECMP は、ほとんどのルーティングプロトコルと組み合わせて使用できます。これは、ルーティングプロトコルはホップごとに決定され、1 つのルータに限定されるためです。複数のパスでトラフィックをロードバランシングすることにより、帯域幅が大幅に増加する可能性があります。

次のコマンドは、ルーティングプロトコルによって送信可能な等コストパスの最大数を設定します。

```
config
  context context_name
    ip routing maximum-paths [ max_num ]
```

注：

- *max_num* は、1 ～ 10（*18.2* よりも前のリリース）または 1 ～ 32（リリース *18.2+*）の整数です。
- 「設定の確認と保存」の章の説明に従って、設定を保存します。

BGP-4 ルーティング

ボーダー ゲートウェイ プロトコル 4（BGP-4）ルーティングプロトコルは、コンテキストレベルで実装されている BGP ルータプロセスを介してサポートされています。

ボーダー ゲートウェイ プロトコル（BGP）は、AS 間ルーティングプロトコルです。自律システム（AS）は、内部ゲートウェイプロトコルと AS 内のパケットをルーティングするための共通メトリックを使用する 1 つの技術的な管理下にあるルータのセットです。ルータのセットは、外部ゲートウェイプロトコルを使用して、他の自律システムにパケットをルーティングします。

BGP は TCP を介して実行されます。これにより、BGP プロトコルによる明示的な更新のフラグメント化、再送信、確認応答、およびシーケンシング情報の実装が不要になります。BGP 独自の認証メカニズムに加えて、TCP で使用される認証方式が使用される場合があります。

BGP ルータは、他の BGP ルータとネットワーク到達可能性情報を交換します。この情報によって、ルートがフィルタ処理され、AS レベルのポリシー決定が適用される AS 接続の実像がつかめます。

BGP-4 には、クラスレスドメイン間ルーティングが備わっています。これには、IP プレフィックスをアドバタイズするためのサポートが含まれており、BGP 内のネットワーククラスという概念を排除します。また、BGP 4 では、AS パスの集約を含むルートの集約も可能です。

重要　VPC-DI では、IPv6 プレフィックス長が /12 未満と /64 ～ /128 の範囲の BGP ルートはサポートされていません。

BGP サポートの概要

モバイルデバイスは、ホームエージェント（HA）を通じてインターネットと通信します。HA は、設定されたアドレスプールから IP アドレスをモバイルノードに割り当てます。これらのアドレスは、ダイナミックルーティングを保証するために、IP ルーティングプロトコルを使用してインターネットルータにもアドバタイズされます。BGP-4 プロトコルは、シャーシ間セッションリカバリ（ICSR）をサポートするためのルーティングを備えた HA とインターネットルータ間のモニタリングメカニズムとして使用されます。（詳細については、「シャーシ間セッションリカバリ」を参照してください）。

BGP-4 プロトコルのサポートの目的は、ルーティング要件を満たし、インターネットルータとの通信をモニタすることです。BGP-4 は、サブスクライバサービスの中断を防ぐために、アクティブからスタンバイへのスイッチオーバーをトリガーする場合があります。

サポートされる BGP-4 機能は次のとおりです。

- 外部ボーダー ゲートウェイ プロトコル（EBGP）マルチホップ
- インバウンドおよびアウトバウンドルートのルートフィルタリング

- ルート再配布とルートマップ
- ルートマップでの BGP コミュニティおよび BGP 拡張コミュニティのサポート
- IPv4 および IPv6（IBGP ピア）のローカルプリファレンス

IP プールルートとループバックルートは、次の方法で BGP ドメインでアドバタイズされます。

- BGP コンフィギュレーション モードの **redistribution** コマンドを使用すると、接続されているすべてのルートまたはその一部が BGP ドメインに再配布されます。（IP プールとループバックルートは、接続されたルートとして IP ルーティングテーブルに存在します）。**network routemap** コマンドを使用すると、多くの BGP 属性を柔軟に変更できます。
- BGP コンフィギュレーションモードの **network** コマンドを使用すると、接続されたルートは BGP ドメインにアドバタイズされるように明示的に設定されます。**network routemap** コマンドを使用すると、多くの BGP 属性を柔軟に変更できます。これらのコマンドの詳細については、『*Command Line Interface Reference*』の「*BGP Configuration Mode Commands*」の章を参照してください。

重要 処理カードの障害、移行、クラッシュ、または処理カードの削除のために BGP タスクが再起動した場合は、すべてのピアリングセッションとルート情報が失われます。

BGP の設定

この項では、システムでの基本的な BGP ルーティングサポートを設定し、有効にする方法について説明します。

```
config
   context context_name
      router bgp AS_number
         neighbor ip_address remote-as AS_num
```

注：

- コンテキストごとに最大 64 の BGP ピアがサポートされます。
- 「設定の確認と保存」の章の説明に従って、設定を保存します。

BGP へのルートの再配布（オプション）

ルートを BGP に再配布するということは、ルートタイプ、ルートマップ内のルールなど、指定された基準を満たす別のプロトコルからのすべてのルートが、BGP プロトコルを使用してすべての BGP エリアに再配布されることを意味します。この設定は、オプションです。

```
config
   context context_name
      router bgp as_number
```

```
      redistribute bgp { bgp | connected | static } [ metric  metric_value
] [ metric-type { 1 | 2 } ] [ route-map  route_map_name ]
```

注：

- 再配布オプションは、connected、ospf、rip、または static です。**redistribute** コマンドの詳細については、『*Command Line Interface Reference*』の「*Border Gateway Protocol Configuration Mode Commands*」の章を参照してください。
- コンテキストごとに最大 64 の ルートマップがサポートされます。
- 「設定の確認と保存」の章の説明に従って、設定を保存します。

BGP コミュニティと BGP 拡張コミュニティ

BGP コミュニティまたは拡張コミュニティ（ルートターゲット）に基づくルートフィルタリングは、CLI ルートマップのコンフィギュレーション モードのコマンドを介して設定します。

BGP コミュニティ

BGP コミュニティの設定

BGP コミュニティは、共通するいくつかの属性を共有する接続先のグループです。各宛先は複数のコミュニティに属します。自律システム管理者は、接続先が属するコミュニティを定義します。

BGP コミュニティは、コンテキスト コンフィギュレーション モードのコマンドを使用して設定します。

```
config
   context context_name
      ip community-list { named named_list | standard  identifier } { deny
| permit } { internet | local-AS | no-advertise | no-export | value
AS-community_number AS-community_number AS-community_number ...}
       { internet | local-AS | no-advertise | no-export | value
AS-community_number AS-community_number AS-community_number ...}
       { internet | local-AS | no-advertise | no-export | value
AS-community_number AS-community_number AS-community_number ...}
```

次の BGP コミュニティの接続先を許可または拒否できます。

- **internet**：このルートをインターネット コミュニティと、そのコミュニティに属するあらゆるルータにアドバタイズします。
- **local-AS**：パケットがローカルの自律システム（AS）外部へ送信されないようにするために、コンフェデレーション シナリオで使用します。
- **no-advertise**：どの BGP ピア（内部または外部）にもこのルートをアドバタイズしません。
- **no-export**：外部 BGP（eBGP）ピアへアドバタイズしません。このルートは AS 内に留められます。
- **value** *community_number*：AS:NN 形式のコミュニティ文字列を指定します。ここで、 AS は 2 バイトの AS コミュニティ 16 進数、NN は 2 バイトの 16 進数です（1 ～ 11 文字）。

各コミュニティには、複数の接続先と AS コミュニティ番号を入力できます。詳細については、『*Command Line Interface Reference*』を参照してください。

複数のコミュニティリストエントリをコミュニティリストに付加するには、さまざまなコミュニティストリングに対して複数の permit 句または deny 句を追加します。最大 64 のコミュニティリストをコンテキストで設定できます。

コミュニティ属性の設定

BGP コミュニティ属性は、ルートマップ内の **set community** コマンドを介して設定します。

```
config
   context context_name
       route-map map_name { deny | permit } sequence_number
          set community [additive]{ internet | local-AS | no-advertise
| no-export | none | value AS-community_number AS-community_number
AS-community_number ...}
          { internet | local-AS | no-advertise | no-export | none | value
 AS-community_number AS-community_number AS-community_number ... }
          { internet | local-AS | no-advertise | no-export | none | value
 AS-community_number AS-community_number AS-community_number ... }
```

additive オプションを指定すると、複数の接続先および AS コミュニティ番号を入力できます。詳細については、『*Command Line Interface Reference*』を参照してください。

BGP コミュニティを介したフィルタリング

BGP コミュニティに基づいてルートをフィルタ処理するには、ルートマップで **match** 句を設定します。コマンドシーケンスは次のとおりです。

```
config
   context context_name
       route-map map_name { deny | permit } sequence_number
          match community { named  named_list | standard  identifier }
```

BGP 拡張コミュニティ

BGP 拡張コミュニティの設定（ルーターゲット）

BGP 拡張コミュニティは、ルーターゲットを定義します。MPLS VPN は、ルーターゲット（RT）と呼ばれる 64 ビットの拡張コミュニティ属性を使用します。RT は、適切な情報テーブルへの到達可能性情報の配信を可能にします。

BGP 拡張コミュニティは、コンテキスト コンフィギュレーション モードのコマンドを使用して設定します。

```
config
   context context_name
       ip extcommunity-list { named named_list | standard  identifier } {
deny | permit } rt rt_number rt_number rt_number ...
```

rt_number は AS:NN 形式の文字列としてルートターゲットを指定します。ここで、 AS は 2 バイトの AS コミュニティ 16 進数、NN は 2 バイトの 16 進数です（1 ～ 11 文字）。IP extcommunity リストには複数のルート番号を追加できます。

さまざまな拡張コミュニティ文字列に複数の permit 句または deny 句を追加することで、拡張コミュニティリストに複数の拡張コミュニティリストエントリを付加できます。最大 64 個の拡張コミュニティリストをコンテキスト内に設定できます。

拡張コミュニティ属性の設定

ルートマップ内で **set extcommunity** コマンドを使用して BGP 拡張コミュニティ属性を設定します。

```
config
  context context_name
      route-map map_name { deny | permit } sequence_number
       set extcommunity rt rt_number rt_number rt_number ...
```

rt_number は AS:NN 形式の文字列としてルートターゲットを指定します。ここで、 AS は 2 バイトの AS コミュニティ 16 進数、NN は 2 バイトの 16 進数です（1 ～ 11 文字）。IP extcommunity リストには複数のルート番号を追加できます。

BGP 拡張コミュニティを介したフィルタリング

BGP 拡張コミュニティ（ルートターゲット）に基づいてルートをフィルタ処理するには、ルートマップで **match** 句を設定します。コマンドシーケンスは次のとおりです。

```
config
  context context_name
      route-map map_name { deny | permit }
        [no] match extcommunity { named  named_list | standard  identifier
 }
```

BGP ローカル プリファレンス

BGP ローカルプリファレンス属性は、IBGP ピアにのみ BGP スピーカーによって送信されます。次のコマンドシーケンスを使用してルートマップ内に設定されます。

```
config
  context context_name
      route-map map_name { deny | permit }
         set local-preference pref_number
```

ルートマップのローカルプリファレンスに対応する **match** 句はありません。これは、ローカルプリファレンスがルート選択アルゴリズムで直接使用されるためです。

ICSR および SRP グループ

BGP は、サービス冗長性プロトコル（SRP）を介してリンクされたシャーシ間セッションリカバリ（ICSR）の設定で採用されています。デフォルトでは、ICSR フェールオーバーは、コンテキスト内のすべての BGP ピアがダウンしたときにトリガーされます。

必要に応じて、コンテキスト内に SRP ピアグループを設定できます。ICSR フェールオーバーは、グループ内のすべてのピアで障害が発生した場合に発生します。このオプションは、IPv4 と IPv6 のピアの組み合わせが複数ペアの VLAN に分散され、ピアグループのすべてのメンバーによって IPv4 または IPv6 の接続が失われる展開で役立ちます。

詳細については、このガイドの「シャーシ間セッションリカバリ」、および『*Command Line Interface Reference*』の「*Service Redundancy Protocol Configuration Mode Commands*」の章にある **monitor bgp**、**monitor diameter** および **monitor authentication-probe** コマンドの説明を参照してください。

スタンバイ ICSR シャーシからの BGP ルートのアドバタイジング

SRP コンフィギュレーション モード コマンドは、スタンバイ状態で ICSR シャーシからの BGP ルートのアドバタイジングを有効にします。このコマンドとそのキーワードを使用すると、オペレータは、光トランスポート ネットワーク ジェネレーション ネクスト（OTNGN）で BGP プレフィックス独立コンバージェンス（PIC）を展開することで、より高速なネットワーク コンバージェンスを利用することができます。

BGP PIC は、アグレッシブな ICSR 障害検出タイマーを安全に設定できるネットワーク コンバージェンスを向上させることを目的としています。

```
configure
   context context_name
      service-redundancy-protocol
       advertise-routes-in-standby-state [ hold-off-time hold-off-time ]
[ reset-bfd-nbrs bfd-down-time ]
       end
```

注：

- **hold-off-time** *hold-off-time* は、タイマーが期限切れになるまで、BGP ルートのアドバタイジングを遅延させます。1 ～ 300 の整数として *hold-off-time* を秒単位で指定します。
- BFD をリセットした後、**reset-bfd-nbrs** *bfd-down-time* は、BFD セッションを設定されたミリ秒数の間停止して、ネットワーク コンバージェンスを向上させます。*bfd-down-time* を 50 ～ 120000 の整数として指定します。

ICSR の設定可能な BGP ルートのアドバタイズメント間隔

デフォルトでは、MinRtAdvInterval が各ピアに設定されます。その値は iBGP ピアの場合は 5 秒、eBGP ピアの場合は 30 秒です。オペレータは、 **neighbor** *identifier* **advertisement-interval** コマンドを使用して、デフォルトの間隔をグローバルに変更できます。

BGP advertisement-interval は、アドレスファミリごとに個別に設定することもできます。設定されている場合、この値は、そのアドレスファミリにおけるピアのデフォルトの advertisement-interval のみをオーバーライドします。BGP は、AFI/SAFI に設定された advertisement-interval に基づいて、AFI/SAFI ごとにルート更新メッセージを送信します。AFI/SAFI advertisement-interval が設定されていない場合は、ピアベースのデフォルトの advertisement-interval が使用されます。

ICSR 設定では、この機能を使用して、ルートアドバタイズメントを高速化し、ネットワークコンバージェンス時間を短縮することができます。

timers bgp icsr-aggr-advertisement-interval コマンドは、BGP アドレスファミリ（VPNv4/VPNv6）コンフィギュレーション モードと BGP アドレスファミリ（VRF）コンフィギュレーション モードの両方で使用できます。

```
configure
   context context_name
      router bgp as_number
         address-family { ipv4 | ipv6 | vpnv4 | vpnv6 }
            timers bgp icsr-aggr-advertisement-interval seconds
```

注：

- *seconds*：0 ～ 30 の整数で秒数を設定します。デフォルト：0。

BGP CLI 設定コマンド

次の表に、さまざまな BGP パラメータの設定をサポートする BGP コンフィギュレーション モードの CLI コマンドを示します。詳細については、『*Command Line Interface Reference*』の「*BGP Configuration Mode Commands*」の章を参照してください。

.

```
configure
   context context_name
      router bgp as_number
```

表 *1: BGP* コンフィギュレーション モードの *CLI* コマンド

bgp コマンド	説明
accept-zero-as-rd	管理サブフィールドを持つルート識別子（RD）値（AS 番号 0）がある VPN プレフィックスを受け入れるように設定します。
address-family { ipv4 | ipv6 }	IPv4 または IPv6 アドレスファミリ設定モードを開始します。
address-family { vpnv4 | vpnv6 }	VPNv4 または VPNv6 アドレスファミリ設定モードを開始します。
bgp graceful-restart { restart-time *rest_time* **| stalepath-time** *stale_time* | **update-delay** *delay*	グレースフルリスタートに関する BGP 固有のパラメータを定義します。
description *text*	この設定の説明テキストを入力できます。
distance { admin *distance* **prefix** *prefix_addr* **[route-access-list** *list_name* **] | bgp external** *ebgp_dist* **internal** *ibgp_dist* local *local_dist* **}**	ルートのアドミニストレーティブ ディスタンスを定義します。アドミニストレーティブディスタンスは、特定のルートまたはタイプルートのデフォルトの優先順位です。

bgp コマンド	説明
enforce-first-as	Exterior Border Gateway Protocol（eBGP）ルートの最初の AS を適用します。
ip vrf *vrf_name*	事前に設定された IP VRF コンテキストインスタンスを BGP ASN に追加し、BGP 属性と関連パラメータを VRF に設定します。
maximum-paths { ebgp *max_num* **| ibgp** *max_num* **}**	複数のパスを介したパケットの転送を有効にし、ネイバー間の外部 BGP（eBGP）パスまたは内部 BGP（iBGP）パスの最大数を指定します。
neighbor *ip_address* **{ activate | advertisement-interval** *adv_time* **| capability graceful-restart | default-originate [route-map** *map_name* **] | distribute-list** *dist_list*{ **in | out } | ebgp-multihop [max-hop number] | encrypted password** e*ncrypted_password* **| fall-over bfd [multihop] | filter-list filt_list { in | out } | max-prefix** *max_num* **[threshold** *thresh_percent* **] [warning-only] | next-hop-self | password** *password* **| remoteas** *AS_num* **| remove-private-AS | restart-time** *rest_time* **| route-map** *map_name* **{ in | out } | send-community { both | extended | standard } | shutdown | srp-activated-soft-clear | timers { [connect-interval** *conn_time* **] | [keepalive-interval** *keep_time* **holdtimeinterval** *hold_time* **] } | update-source** *ip_address* **| weight** *value* **}**	非ブロードキャスト ネットワークに相互接続する BGP ルータを設定します。他のパラメータを設定する前に、ネイバーに対してリモート AS 番号を指定する必要があることに注意してください。 注：**advertisement-interval** は、そのアドレスファミリに有効になるように、アドレスファミリに対して明示的に設定する必要があります。デフォルトでは、IPv4 アドレスファミリにのみ適用されます。**address-family** コマンド使用してアドレスファミリを指定します。その後で、アドレス ファミリ コンフィギュレーション モードでネイバーアドバタイズメント間隔を設定できます。
network *ip_address/mask* **[route-map** *map_name* **]**	BGP を介してアナウンスするネットワークを指定します。
redistribute { connected | ospf | rip | static } [route-map *map_name* **]**	BGP を介して別のプロトコルから BGP にルートを再配布します。
router-id *ip_address*	設定されたルータ ID を上書きし、BGP ピアをリセットします。
scan-time *time*	BGP バックグラウンドスキャナの間隔を秒単位で設定します。BGP は、インストールされているルートのネクストホップをモニタして、ネクストホップの到達可能性を確認し、BGP ベストパスループを選択、インストール、および検証します。
timers bgp keepalive-interval *interval* **holdtime-interval** *time* **[min-peer-holdtimeinterval** *time* **]**	BGP ルーティングタイマーを設定します。

BGP 設定パラメータの確認

BGP ルータの設定を確認するには、次のコマンドを使用して、画面出力で「**router BGP**」というラベルの付いたセクションを探します。

show config context *ctxt_name* **[verbose]**

BGP ピアの制限

機能の概要と変更履歴

要約データ

該当製品または機能エリア	すべて
該当プラットフォーム	VPC - DI
機能のデフォルト	無効：設定が必要
このリリースでの関連する変更点	N/A
関連資料	• *Command Line Interface Reference* • *Statistics and Counters Reference* • *VPC-DI* システム管理ガイド

マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	21.8

機能説明

アーキテクチャを備えた Cisco 仮想化パケットコア分散型インスタンス（VPC-DI）では、パケット処理カード（逆多重化 SF カードを含むセッション機能（SF））間での BGP ピアリングに柔軟に対応できます。

SDN の「Contrail」モデルに基づく展開手順では、各パケット処理カードにはコンピューティングノード内に vRouter があります。このモデルでは、現在の柔軟な BGP ピアリングスキームを使用して、BGP の設定をこれらの vRouter それぞれに実装する必要があります。そのため、ネットワーク内に多数の SF カードがある場合は、サービスプロバイダーにとっての課題となります。設定が必要な回線数は拡張性の問題をもたらします。

この問題を解決するため、BGP ピア制限機能が導入されました。この機能は BGP ピアリングを VPC-DI アーキテクチャ内の **2** 枚の SF カードのみに制限します。この機能では、ルーティングテーブルには、その 2 枚の SF カードに対応するルート 2 つのみが存在し、3 番目のルー

トは「ブラックホール」または「ヌル」ルートとなります。新しいルートが最長プレフィックスに一致するルートであることを確認するために、ホストアドレスのみ（/32 ビットマスク）のプロビジョニングのみが適用されます。これにより、設定の量とルーティングテーブルのサイズが大幅に削減されます。

機能の仕組み

この機能は、**ip route kernel** コマンドを使用して実装されます。設定すると、BGP ピアリングは特別なルートを持つ SF カード 2 枚にのみ制限されます。

blackhole キーワードを設定すると、カーネル ルーティング エンジンが有効になり、ノードから発信されるパケットをブロックまたはドロップできます。これは、どのインターフェイスにも限定されず、デフォルトではワイルドカード インターフェイスになります。

BGP ピア制限機能の設定の詳細については、「BGP ピア制限の設定」の項を参照してください。

制限事項

- この機能のサポートは、コンテキストレベルのみに制限されています。
- VRF レベルではサポートされていません。
- この機能は、IPv4 でのみサポートされています。

BGP ピア制限の設定

次の項では、この機能を有効または無効にするための設定コマンドについて説明します。

パケット処理カードのルート設定

次の CLI コマンドを使用して、コンテキスト コンフィギュレーションで定義されている任意の 2 つのパケット処理インターフェイス（SF カード）に特別な（スタティック）ルートを追加します。

```
configure
  context context_name
    [ no ] ip route kernel ip_address/ip_address_mask_combo
egress_intrfc_name cost number
    end
```

注：

- **no**：追加されたルートを削除します。
- **kernel**：カーネルのルーティングテーブルのオプションでスタティックルートを許可します。
- **ip_address/ip_address_mask_combo**：ルートが適用される IP アドレスを示すために、結合された IP アドレスのサブネットマスクビットを指定します。*ip_address_mask_combo* は、CIDR 表記を使用して指定する必要があります。これは、IP アドレスが IPv4 ドット付き

10進表記を使用して指定されたもので、マスクビットは数値（サブネットマスクのビット数）です。

- *egress_intrfc_name*： 1 ～ 79 文字の英数字の文字列として、既存の出力インターフェイスの名前を指定します。
- **cost***number*：次のゲートウェイへのホップ数を定義します。コストは 0 ～ 255 の整数にする必要があります。255が最も高価です。デフォルトは 0 です。
- この機能は、デフォルトでディセーブルになっています。

ブラックホールルートの設定

ノードから発信されるパケットをブロックまたはドロップするには、次のCLIコマンドを使用します。

```
configure
  context context_name
    [ no ] ip route kernel ip_address/ip_address_mask_combo
egress_intrfc_name cost number blackhole
    end
```

注：

- **no**：追加されたルートを削除します。
- **kernel**：カーネルのルーティングテーブルのオプションでスタティックルートを許可します。
- **ip_address/ip_address_mask_combo**：ルートが適用されるIPアドレスを示すために、結合されたIPアドレスのサブネットマスクビットを指定します。*ip_address_mask_combo* は、CIDR 表記を使用して指定する必要があります。これは、IP アドレスが IPv4 ドット付き10進表記を使用して指定されたもので、マスクビットは数値（サブネットマスクのビット数）です。
- *egress_intrfc_name*： 1 ～ 79 文字の英数字の文字列として、既存の出力インターフェイスの名前を指定します。デフォルトは「*」（ワイルドカードインターフェイス）です。
- **cost***number*：次のゲートウェイへのホップ数を定義します。コストは 0 ～ 255 の整数にする必要があります。255が最も高価です。デフォルトは 0 です。
- **blackhole**：パケットをブロックまたはドロップするため、カーネルにインストールするブラックホールルートを定義します。
- この機能は、デフォルトでディセーブルになっています。

モニタリングおよびトラブルシューティング

この項では、機能のモニタリングとトラブルシューティングのサポートに使用できるCLIコマンドに関する情報を提供します。

コマンドや出力の表示

この項では、この機能のサポートにおける show コマンドやその出力に関する情報を提供します。

show ip route

この show command CLI は、任意の 2 つのパケット処理インターフェイス（SF カード）にスタティックルートが追加されると、次の新しいフィールドの値を挿入します。

カーネル専用

双方向フォワーディング検出

Bidirectional Forwarding Detection（BFD）は、リンクによって接続されている 2 つの転送エンジン間の障害を検出するために使用されるネットワークプロトコルです。BFD は、特定のリンクを介した 2 つのエンドポイント間でセッションを確立します。2 つのシステム間に複数のリンクが存在する場合、それぞれをモニタするために複数の BFD セッションが確立される場合があります。セッションは 3 ウェイハンドシェイクによって確立され、同じ方法で切断されます。セッションで認証が有効になっている場合があります。簡易パスワード、MD5、または SHA1 認証を選択できます。

重要 VPC-DI では、IPv6 プレフィックス長が /12 未満で、/64 と/128 の範囲の BFD ルートはサポートされていません。

BFD サポートの概要

BFD には検出メカニズムがありません。エンドポイント間にセッションを明示的に設定する必要があります。基盤となるさまざまな転送メカニズムやレイヤの多くで BFD が使用されている可能性があり、また、それらすべては独立して動作しています。そのため、使用する転送が何であっても、BFD をカプセル化する必要があります。

OSPF や IS-IS などの何らかの形式の隣接セットアップをサポートするプロトコルは、BFD セッションの起動ストラップにも使用できます。これらのプロトコルは、プロトコル独自のキープアライブメカニズムを使用して通常は可能になるよりも迅速に障害が発生しているリンクの通知を受信するために BFD を使用できます。

非同期モードでは、エンドポイントの両方が定期的の Hello パケットを相互に送信しています。これらのパケットを複数回受信しない場合は、セッションがダウンしていると見なされます。

エコーがアクティブになると、エコーパケットのストリームがもう一方のエンドポイントに送信され、その後にそれらが送信元に転送されます。エコーは、**bfd protocol** コマンドを使用してグローバルに有効にすることも、インターフェイスごとに個別に有効または無効にすることもできます。この機能は、リモートシステム上の転送パスのテストに使用されます。

システムは、スタティックルーティング、または BGP ルーティングを介して、オプションのエコー機能を使用した非同期モードで BFD をサポートしています。

重要 BFD を機能させるには、ASR 5500 ではいずれかのパケット処理カードをデマルチプレクサカードとして設定する必要があります。詳細については、「システム設定」の章の「逆多重化カードの設定」の項を参照してください。

BFD の設定

この項では、システムでの基本的な BFD ルーティングプロトコルのサポートを設定し、有効にする方法について説明します。

BFD プロトコルの設定に影響する要因はいくつかあります。

BFD コンテキストの設定

```
config
  context context_name
    bfd-protocol
    [ bfd echo ]
    exit
```

注：

- エコー機能は、このコンテキストのすべてのインターフェイスに対して必要に応じて有効にすることができます。
- コンテキストごとに 16 BFD セッション、およびシャーシごとに 64 BFD セッション。

スタティックルートの IPv4 BFD の設定

インターフェイス上で BFD を有効にします。

```
config
   context bfd_context_name
      interface if_name
         ip address ipv4_address ipv4_mask
         bfd interval interval_value min_rx rx_value multiplier multiplier_value
         [ bfd echo ]
      exit
```

BFD スタティックルートを設定します。

```
ip route static bfd if_name ipv4_gw_address
```

スタティックルートを追加します。

```
ip route ipv4_address ipv4_mask
ip route ipv4_address ipv4_mask
```

スタティックルートの IPv6 BFD の設定

インターフェイスでの BFD を有効にします。

```
config
   context bfd_context_name
   interface if_name
      ipv6 address ipv6_address ipv6_mask
         bfd interval interval_value min_rx rx_value multiplier multiplier_value
         [ bfd echo ]
         exit
```

BFD スタティックルートを設定します。

```
ipv6 route static bfd if_name ipv6_gw_address
```

スタティックルートを追加します。

```
ipv6 route ipv6_address ipv6_mask
ipv6 route ipv6_address ipv6_mask
```

重要　ASR 5500 では、IPv6 プレフィックス長が /12 未満で、/64 ～ /128 の範囲のスタティックルートはサポートされていません。

シングルホップ用の BFD の設定

インターフェイス上で BFD を有効にします。

```
config
   context bfd_context_name
      interface if_name
         ip address ipv4_address ipv4_mask
         ipv6 address ipv6_address ipv6_mask
         bfd interval interval_value min_rx rx_value multiplier multiplier_value
         [ bfd echo ]
         exit
```

BGP ネイバー上で BFD を有効にします。詳細については、BGP ネイバーとコンテキストの関連付け （24 ページ）を参照してください。

OSPF ネイバー上で BFD を有効にします。詳細については、OSPF ネイバーとコンテキストの関連付け （25 ページ）を参照してください。

重要 ASR 5500 では、IPv6 プレフィックス長が /12 未満で、/64 ～ /128 の範囲のルートはサポートされていません。

マルチホップ BFD の設定

インターフェイス上で BFD を有効にします。

```
config
   context bfd_context_name
      interface if_name
         ip address ipv4_address ipv4_mask
         ipv6 address ipv6_address ipv6_mask
         bfd interval interval_value min_rx rx_value multiplier multiplier_value
         [ bfd echo ]
         exit
```

マルチホップ BFD セッションを設定します。

```
      bfd-protocol
        bfd multihop peer  destination-address interval interval-value multiplier
 multiplier-value
```

BGP ネイバー上で BFD を有効にします。詳細については、BGP ネイバーとコンテキストの関連付け （24 ページ）を参照してください。

BFD の拡張性

上記のいずれかの方法を使用してアクティブな BFD セッションを設定し、アクティブインターフェイスの設定時に同じ BFD ネイバーを使用します。詳細については、BFD ネイバーグループと BFD プロトコルの関連付け （25 ページ）を参照してください。

```
      bfd-protocol
         bfd nbr-group-name grp_name active-if-name if_name nexthop_address
```

1 つ以上のパッシブインターフェイスに同じ BFD の結果を適用します。

```
         bfd nbr-group-name grp_name passive-if-name if_name nexthop_address
         bfd nbr-group-name grp_name passive-if-name if_name nexthop_address
```

BGP ネイバーとコンテキストの関連付け

```
config
   context context_name
      router bgp AS_number
         neighbor neighbor_ip-address remote-as rem_AS_number
```

```
neighbor neighbor_ip-address ebgp-multihop max-hop max_hops
neighbor neighbor_ip-address update-source update-src_ip-address
neighbor neighbor_ip-address failover bfd [ multihop ]
```

注：

- ネイバーを追加するには、このシーケンスを繰り返します。

OSPF ネイバーとコンテキストの関連付け

```
config
  context context_name
    router ospf
      neighbor neighbor_ip-address
```

注：

- ネイバーを追加するには、このシーケンスを繰り返します。

BFD ネイバーグループと BFD プロトコルの関連付け

```
config
  context context_name
    bfd-protocol
      bfd nbr-group-name grp_name active-if-name if_name nexthop_address
      bfd nbr-group-name grp_name passive-if-name if_name nexthop_address
```

OSPF インターフェイスでの BFD の有効化

すべての OSPF インターフェイス

```
config
  context context_name
    router ospf
      bfd-all-interfaces
```

特定の OSPF インターフェイス

```
config
  context context_name
    interface interface_name
      broadcast
        ip ospf bfd
```

ICSR の BFD 接続のモニタリング

ICSR 設定では、次のコマンドシーケンスは、指定されたコンテキストのプライマリシャーシと BFD ネイバーとの間の接続のモニタリングを開始します。接続が切断されると、スタンバイシャーシがアクティブになります。

```
config
  context context_name
```

```
    service-redundancy-protocol
      monitor bfd context context_name { ipv4_address | ipv6_address } {
chassis-to-chassis | chassis-to-router }
```

注：

- *ipv4 _address | ipv6_address* は、モニタする BFD ネイバーの IP アドレスを定義します。これは、IPv4 ドット付き 10 進表記または IPv6 コロン区切り 16 進表記を使用して入力します。
- **chassis-to-chassis** は、非 SRP リンク上のプライマリシャーシとバックアップシャーシの間で BFD を実行できるようにします。
- **chassis-to-router** は、シャーシとルータの間で BFD を実行できるようにします。

設定の保存

「設定の確認と保存」の章の説明に従って、設定を保存します。

ICSR のシャーシ間 BFD のモニタリング

オペレータは、ICSR スイッチオーバー時にルートをより迅速にアドバタイズするように BFD を設定できます。このソリューションは、スタンバイ ICSR シャーシからの BGP ルートのアドバタイズを可能にする機能を補完します。VoLTE の実装時に ICSR 設定でより積極的な障害検出と回復をサポートすることを全体的な目標としています。

ICSR 設定でシャーシ間 BFD モニタリングを行うには、次の機能を設定する必要があります。

- プライマリシャーシ BFD モニタリングの有効化 （26 ページ）。
- ICSR Dead 間隔を無視する BFD の設定 （26 ページ）。
- ICSR スイッチオーバー ガード タイマー の設定 （27 ページ）。
- BFD マルチホップ フォールオーバーの有効化 （28 ページ）。
- スタンバイ ICSR シャーシからの BGP ルートのアドバタイジングの有効化 （29 ページ）。

プライマリシャーシ BFD モニタリングの有効化

プライマリシャーシと指定された BFD ネイバー間の接続をモニタリングできるようにする必要があります。接続が切断されると、スタンバイシャーシがアクティブになります。詳細については、「ICSR の BFD 接続のモニタリング （25 ページ）」を参照してください。

ICSR Dead 間隔を無視する BFD の設定

SRP コンフィギュレーションモード **bfd-mon-ignore-dead-interval** コマンドを実行すると、スタンバイ ICSR シャーシは dead 間隔を無視し、すべての BFD シャーシ間モニタが失敗するまでスタンバイ状態のままになります。

この機能は、BFD シャーシ間モニタリングとの関連付けで有効にして、よりアグレッシブな ICSR 障害検出時間をサポートします。

```
configure
   context context_name
```

```
service-redundancy-protocol variable
   bfd-mon-ignore-dead-interval
   end
```

ICSR スイッチオーバー ガード タイマー の設定

SRP コンフィギュレーション モードの **guard timer** コマンドは、SRP サービスのモニタリングのための edundancy-guard-period と monitor-damping-period を設定します。

これらのガードタイマーを使用して、カードの再起動およびタスクの再起動などのローカルな障害により、中断する可能性のある ICSR イベントを発生させることがないようにします。

```
configure
   context context_name
      service-redundancy-protocol variable
         guard-timer { aaa-switchover-timers { damping-period seconds |
guard-period seconds } | diameter-switchover-timers { damping-period
seconds | guard-period seconds } | srp-redundancy-timers { aaa  {
damping-period seconds | guard-period seconds } | bgp { damping-period
seconds | guard-period seconds } | diam  { damping-period seconds |
guard-period seconds } }
         end
```

注：

- **aaa-switchover-timers**：ネットワークのコンバージ中に AAA の障害（post ICSR switchover）が発生した場合に、バックツーバックの ICSR スイッチオーバーを防止するタイマーを設定します。
  - **damping-period**：guard-period 内のモニタリング障害による ICSR スイッチオーバーをトリガーする遅延時間を設定します。
  - **guard-period**：local-failure-recovery と network-convergence のタイマーを設定します。
- **diameter-switchover-timers**：ネットワークのコンバージ中に Diameter の障害（post ICSR switchover）が発生した場合に、バックツーバックの ICSR スイッチオーバーを防止するタイマーを設定します。
  - **damping-period**：guard-period 内のモニタリング障害による ICSR スイッチオーバーをトリガーする遅延時間を設定します。
  - **guard-period**：local-failure-recovery と network-convergence のタイマーを設定します。
- **srp-redundancy-timer**：システムがローカルの card-reboot/critical-task-restart の障害からリカバリされている間に、ICSR スイッチオーバーを防止するタイマーを設定します。
  - **aaa**：ローカル障害に続いて AAA のモニタリングの障害が発生する。
  - **bgp**：ローカル障害に続いて BGP のモニタリングの障害が発生する。
  - **diam**：ローカル障害に続いて Diameter のモニタリングの障害が発生する。

BFD マルチホップ フォールオーバーの有効化

コンテキスト コンフィギュレーション モードの ip route コマンドと ipv6 route コマンドでの **fall-over bfd multihop** *mhsess_name* キーワードは、指定されたマルチホップセッションのフォールオーバー BFD 機能を有効にします。**fall-over bfd** オプションは、BFD を使用してネイバーの到達可能性と活性度をモニタします。有効にすると、BFD が障害を通知した場合にセッションを切断します。

```
configure
   context context_name
      ip route { ip_address/ip_mask | ip_address ip_mask } { gateway_ip_address |
 next-hop next_hop_ip_address | point-to-point | tunnel } egress_intrfc_name [
 cost cost ] [ fall-over bfd multihop mhsess_name ] [ precedence precedence
 ] [ vrf vrf_name [ cost value ] [ fall-over bfd multihop mhsess_name ] [
precedence precedence ] +
      end
```

Ip route コマンドでは、静的マルチホップの BFD ルートを追加することもできます。

```
      ip route static multihop bfd mhbfd_sess_name local_endpt_ipaddr
remote_endpt_ipaddr
```

重要　SNMP トラップは、BFD セッションがアップおよびダウンしたときに生成されます（BFDSessUp と BFDSessDown）。

ip route コマンド

```
configure
   context context_name
      ip route { ip_address/ip_mask | ip_address ip_mask } { gateway_ip_address |
 next-hop next_hop_ip_address | point-to-point | tunnel } egress_intrfc_name [
 cost cost ] [ fall-over bfd multihop mhsess_name ] [ precedence precedence
 ] [ vrf vrf_name [ cost value ] [ fall-over bfd multihop mhsess_name ] [
precedence precedence ] +
      end
```

Ip route コマンドでは、静的マルチホップの BFD ルートを追加することもできます。

```
      ip route static multihop bfd mhbfd_sess_name local_endpt_ipaddr
remote_endpt_ipaddr
```

ip routev6 コマンド

```
configure
   context context_name
      ipv6 route ipv6_address/prefix_length { interface name | next-hop
ipv6_address interface name } [ cost  cost] [ fall-over bfd multihop mhsess_name
 ] [ precedence precedence ] [ vrf vrf_name [ cost value ] [ fall-over bfd
multihop mhsess_name ] [ precedence precedence ]
      end
```

ipv6 route コマンドでは、スタティックマルチホップ BFD ルートも追加できるようになりました。

```
ipv6 route static multihop bfd mhbfd_sess_name local_endpt_ipv6addr
remote_endpt_ipv6addr
```

BFD 間隔の調整

ネットワーク展開のコンバージェンス要件を満たすため、BFDパケット間の送信間隔を設定します（ミリ秒単位）。

```
configure
   context context_name
      interface interface_name broadcast
         bfd interval interval_num  min_rx milliseconds multiplier value
         end
```

注：

- *milliseconds* は、50 ～ 10,000 の整数です（デフォルトは 50）

スタンバイ ICSR シャーシからの BGP ルートのアドバタイジングの有効化

この機能の設定については、スタンバイ ICSR シャーシからの BGP ルートのアドバタイジング（15 ページ） をご覧ください。

設定の保存

「設定の確認と保存」の章の説明に従って、設定を保存します。

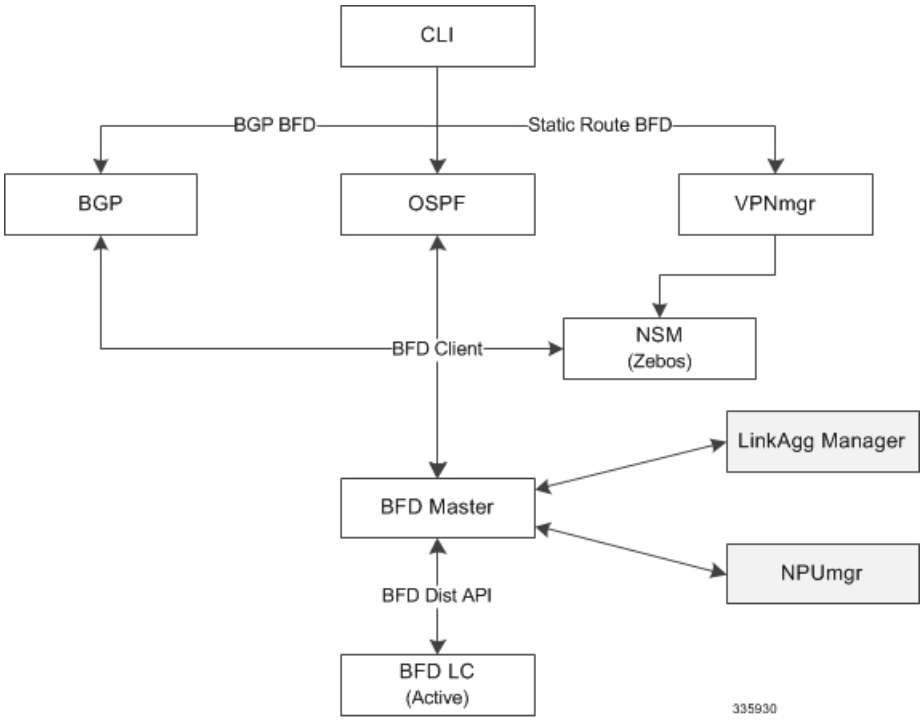
リンク集約メンバーリンクの BFD サポート

メンバーリンクベースの BFD は、LACP よりも高速で個々のリンク障害を検出し、単一のメンバーリンク障害の結果として、セッション全体またはトラフィックのダウン期間を短縮します。

概要

BFD コンフィギュレーション モードの CLI コマンドは、linkagg タスクとの BFD の連携動作を設定します。セッションが設定されると、BFD はメンバリンクごとに BFD セッションを作成し、各 linkagg メンバーリンクでパケットの送信を開始します。メンバリンクの BFD セッションに障害が発生すると、StarOS は linkagg タスクに障害を通知します。

図 *1 : BFD* の連携動作



スロット番号を使用して linkagg-peer を定義する場合は、冗長 LC（ラインカード）スロット用の linkagg peer を設定することもできます。その際、メンバリンク設定でスロットも指定する必要があります。また、スロットを使用せずに linkagg-peer を設定する場合は、スロットを指定してピアを設定する前に、そのピアを削除する必要があります。

重要　RFC 7130 に準拠するため、linkagg インターフェイスごとに許可される IPv4 または IPv6 BFD セッションベースの設定は 1 つだけです。

BFD Linkagg メンバーリンクのサポートの設定

bfd linkagg-peer コマンドは、メンバーリンク BFD を有効にして、BFD リンク集約（linkagg）セッション値を設定します（RFC 7130）。

```
configure
  context context_name
    bfd-protocol
      bfd linkagg-peer linkagg_group_id local-endpt-addr local-endpt_ipaddress
remote-endpt-addr remote_endpt_ipaddress interval tx_interval min_rx rx_interval
multiplier multiplier_value [ slot slot_number ]
      no bfd linkagg-peer linkagg_group_id [ slot slot_number ]
      end
```

注：

- *linkagg_group_id* は、LAG 番号を 1 ～ 255 の整数としてで指定します。
- **local-endpt-addr** *local-endpt_ipaddress* は、IPv4 または IPv6 表記内のマルチホップ BFD セッションの送信元アドレスを指定します。
- **remote-endpt-addr** *remote-endpt_ipaddress* は、IPv4 または IPv6 表記内のマルチホップ BFD セッションのリモートアドレスを指定します。
- **interval** *tx_interval* は、制御パケットの送信間隔を 50 ～ 10000 の整数で（ミリ秒単位）指定します。
- **min_rx** *rx_interval* は、制御パケットの受信間隔を 50 ～ 10000 の整数で（ミリ秒単位）指定します。
- **multiplier** *multiplier_value* は、ホールド時間を計算するために使用する値を 3 ～ 50 の整数で指定します。
- 冗長アクティブ/スタンバイリンク集約の **slot** *slot_number* オプションは、この設定を適用するカードを指定します。

設定の保存

「設定の確認と保存」の章の説明に従って、設定を保存します。

ルーティング情報の表示

現在のコンテキストのルーティング情報を表示するには、次の Exec モードのコマンドのいずれかを実行します。

- **show ip route**：現在のコンテキストにおける IPv4 ルートの情報を表示します。
- **show ipv6 route**：現在のコンテキストにおける ipv6 ルートの情報を表示します。
- **show ip static-route**：現在の contextospf 内の IPv4 スタティックルートの情報のみを表示します。
- **show ip ospf**：現在のコンテキストにおける IPv4 OSPF プロセスの概要情報を表示します。
- **show ipv6 ospf**：現在のコンテキストにおける IPv6 OSPFv3 プロセスの概要情報を表示します。
- **show ip bgp**：IPv4 BGP 情報を表示します。

次に、 **show ip route** コマンドの出力例を示します。

```
[local]host_name# show ip route
"*" indicates the Best or Used route.

 Destination         Nexthop          Protocol   Prec Cost Interface
*44.44.44.0/24       208.230.231.50   static     1    0    local1
*192.168.82.0/24     0.0.0.0          connected  0    0
*192.168.83.0/24     0.0.0.0          connected  0    0
 208.230.231.0/24    0.0.0.0          ospf       110  10   local1
*208.230.231.0/24    0.0.0.0          connected  0    0    local1
Total route count: 5
```