RIB の実装とモニタリング

ルーティング情報ベース（RIB）は、ネットワークのすべてのノード間のルーティングの接続に関する情報を収集して配布したものです。各ルータには、そのルータのルーティング情報を含む RIB を維持します。RIB は、システムで実行されているすべてのルーティング プロトコルでの最良ルートを保存します。

各ルーティング プロトコルは独自の最適ルートのセットを選択し、これらのルートとその属性を RIB に取り込みます。RIB はこれらのルートを格納し、すべてのルーティング プロトコルの中から最適ルートを選択します。これらのルートは転送パケットで使用するために、ラインカードにダウンロードされます。頭字語の RIB は、RIB プロセスと、RIB 内に含まれるルートデータの集合を表すために使用されます。プロトコル内で、ルートはそのプロトコルによって使用されているメトリックに基づいて選択されます。プロトコルは最適なルート（最も低いメトリックまたは結び付けられたメトリック）を RIB にダウンロードします。RIB は、関連付けられているプロトコルのアドミニストレーティブ ディスタンスを比較して、全体的に最適なルートを選択します。

このモジュールでは、ネットワークで RIB を実装およびモニタリングする方法を説明します。

（注） VPNv4、VPNv6 および VPN ルーティング/転送（VRF）のアドレス ファミリは、今後のリリースでサポートされる予定です。

ルーティング テーブルを使用した RIB 設定の確認

ルーティング テーブルの概要と詳細の情報をチェックすることで、RIB が RP 上で実行され、正常に機能していることを確認するために、RIBの設定を確認するには、次の作業を実行します。

手順

ステップ 1 **show route** [**vrf** { *vrf-name* | **all** }] [**afi-all** | **ipv4** | **ipv6**] [**unicast** | **safi-all**] **summary** [**detail**] [**standby**]

例：

```
RP/0/RP0/cpu 0: router# show route summary
```

指定したルーティング テーブルに関するルート サマリー情報を表示します。

- 要約されたデフォルト テーブルは、IPv4 ユニキャスト ルーティング テーブルです。

ステップ 2 **show route** [**vrf** { *vrf-name* | **all** }] [**afi-all** | **ipv4** | **ipv6**] [**unicast** | **safi-all**] [*protocol* [*instance*] | *ip-address mask*] [**standby**] [**detail**]

例：

```
RP/0/RP0/cpu 0: router# show route ipv4 unicast
```

指定したルーティング テーブルに関する詳細なルート情報を表示します。

- このコマンドは、表示を制限するために通常は IP アドレスまたは他のオプション フィルタを使用して発行します。それ以外の場合は、デフォルトの IPv4 ユニキャスト ルーティングテーブルからすべてのルートを表示します。ネットワークの設定に応じて大規模なリストになる可能性があります。

show route best-local コマンドの出力：例

次に、**show route backup** コマンドの出力例を示します。

```
show route backup

 Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
        O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local
 S    172.73.51.0/24 is directly connected, 2d20h, HundredGigE 0/9/0/0
```

```
Backup  O E2 [110/1] via 10.12.12.2, HundredGigE 0/9/0/0
```

ネットワーキングとルーティングの問題の検証

ノード間のルートの動作を検証するには、次のタスクを実行します。

手順

ステップ 1 **show route** [**vrf** { *vrf-name* | **all** }] [**afi-all** | **ipv4** | **ipv6**] [**unicast** | **safi-all**] [*protocol* [*instance*] | *ip-address mask*] [**standby**] [**detail**]

例：

```
RP/0/RP0/cpu 0: router# show route ipv4 unicast 192.168.1.11/8
```

RIB の現在のルートを表示します。

ステップ 2 **show route** [**vrf** { *vrf-name* | **all** }] [**afi-all** | **ipv4** | **ipv6**] [**unicast** | **safi-all**] **backup** [*ip-address*] [**standby**]

例：

```
RP/0/RP0/cpu 0: router# show route ipv4 unicast backup 192.168.1.11/8
```

RIB のバックアップ ルートを表示します。

ステップ 3 **show route** [**vrf** { *vrf-name* | **all** }] [**ipv4** | **ipv6**] [**unicast** | **safi-all**] **best-local** *ip-address* [**standby**]

例：

```
RP/0/RP0/cpu 0: router# show route ipv4 unicast best-local 192.168.1.11/8
```

特定の宛先からの応答パケットに使用する場合に最善のローカル アドレスが表示されます。

ステップ 4 **show route** [**vrf** { *vrf-name* | **all** }] [**afi-all** | **ipv4** | **ipv6**] [**unicast** | **safi-all**] **connected** [**standby**]

例：

```
RP/0/RP0/cpu 0: router# show route ipv4 unicast connected
```

ルーティング テーブルの現在の接続ルートを表示します。

ステップ 5 **show route** [**vrf** { *vrf-name* | **all** }] [**afi-all** | **ipv4** | **ipv6**] [**unicast** | **safi-all**] **local** [*interface*] [**standby**]

例：

```
RP/0/RP0/cpu 0: router# show route ipv4 unicast local
```

ルーティング テーブルの受信エントリのローカル ルートを表示します。

ステップ 6 **show route** [**vrf** { *vrf-name* | **all** }] [**ipv4** | **ipv6**] [**unicast** | **safi-all**] **longer-prefixes** { *ip-address mask* | *ip-address* / *prefix-length* } [**standby**]

例：

```
RP/0/RP0/cpu 0: router# show route ipv4 unicast longer-prefixes 192.168.1.11/8
```

指定のネットワークと指定の数のビットを共有する RIB の現在のルートを表示します。

ステップ 7 **show route** [**vrf** { *vrf-name* | **all** }] [**ipv4** | **ipv6**] [**unicast** | **safi-all**] **next-hop** *ip-address* [**standby**]

例：

```
RP/0/RP0/cpu 0: router# show route ipv4 unicast next-hop 192.168.1.34
```

宛先アドレスまでのネクスト ホップ ゲートウェイまたはホストを表示します。

---

show route コマンドの出力：例

次に、アドレスを指定せずに入力した show route コマンドの出力例を示します。

```
show route

  Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
         O - OSPF, IA - OSPF inter area
         N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
         E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
         i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
         ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
         U - per-user static route, o - ODR, L - local

  Gateway of last resort is 172.23.54.1 to network 0.0.0.0

  C    10.2.210.0/24 is directly connected, 1d21h, Ethernet0/1/0/0
  L    10.2.210.221/32 is directly connected, 1d21h, Ethernet0/1/1/0
  C    172.20.16.0/24 is directly connected, 1d21h, GigabitEthernet 0/4/0/0
  L    172.20.16.1/32 is directly connected, 1d21h, GigabitEthernet 0/4/0/0
  C    10.6.100.0/24 is directly connected, 1d21h, Loopback1
  L    10.6.200.21/32 is directly connected, 1d21h, Loopback0
  S    192.168.40.0/24 [1/0] via 172.20.16.6, 1d21h
```

RIB ネクストホップ ダンプニングのディセーブル化

RIB ネクスト ホップ ダンプニングをディセーブルにするには、次のタスクを実行します。

手順

ステップ 1 **router rib**

例：

```
RP/0/RP0/cpu 0: router# route rib
```

RIB コンフィギュレーション モードを開始します。

ステップ 2 **address-family** { **ipv4** | **ipv6** } **next-hop dampening disable**

例：

```
RP/0/RP0/cpu 0: router(config-rib)# address family ipv4 next-hop dampening disable
```

IPv4 アドレス ファミリのネクスト ホップ ダンプニングをディセーブルにします。

ステップ 3 **commit**

show route next-hop コマンドの出力：例

次に、**show route resolving-next-hop** コマンドの出力例を示します。

```
show route resolving-next-hop 10.0.0.1

Nexthop matches 0.0.0.0/0
  Known via "static", distance 200, metric 0, candidate default path
  Installed Aug 18 00:59:04.448
  Directly connected nexthops

    172.29.52.1, via MgmtEth0/RP1/CPU0/0
      Route metric is 0
```

RCC および LCC オンデマンド スキャンのイネーブル化

ルート整合性チェッカ（RCC）、およびラベル整合性チェッカ（LCC）オンデマンドスキャンをトリガーするには、次の作業を実行します。オンデマンドスキャンは、特定のアドレスファミリ（AFI）で、サブアドレスファミリ（SAFI）、テーブル、および、プレフィックス、VRF、またはテーブルのすべてのプレフィックスに関して実行できます。

手順

ステップ 1 次のいずれかのコマンドを使用します。

- **show rcc** {**ipv4** | **ipv6**} **unicast** [**all**] [*prefix*/*mask*] [**vrf** *vrf-name*]
- **show lcc** {**ipv4** | **ipv6**} **unicast** [**all**] [*prefix*/*mask*] [**vrf** *vrf-name*]

例：

```
RP/0/RP0/cpu 0: router#show rcc ipv6 unicast 2001:DB8::/32 vrf vrf_1
```

または

```
RP/0/RP0/cpu 0: router#show lcc ipv6 unicast 2001:DB8::/32 vrf vrf_1
```

ルート整合性チェッカ（RCC）またはラベル整合性チェッカ（LCC）オンデマンドで実行します。

ステップ 2 次のいずれかのコマンドを使用します。

- **clear rcc** {**ipv4** | **ipv6**} **unicast** [**all**] [*prefix*/*mask*] [**vrf** *vrf-name*] **log**
- **clear lcc** {**ipv4** | **ipv6**} **unicast** [**all**] [*prefix*/*mask*] [**vrf** *vrf-name*] **log**

例：

```
RP/0/RP0/cpu 0: router#clear rcc ipv6 unicast log
```

または

```
RP/0/RP0/cpu 0: router#show lcc ipv6 unicast log
```

以前のスキャンのログをクリアします。

RCC および LCC バックグラウンド スキャンのイネーブル化

ルート整合性チェッカ（RCC）およびラベル整合性チェッカ（LCC）のバックグラウンド スキャンを実行するには、次のタスクを実行します。

手順

ステップ 1 **configure**

ステップ 2 次のいずれかのコマンドを使用します。

- **rcc** {**ipv4** | **ipv6**} **unicast** {**enable** | **period** *milliseconds*}
- **lcc** {**ipv4** | **ipv6**} **unicast** {**enable** | **period** *milliseconds*}

例：

```
RP/0/RP0/cpu 0: router(config)#rcc ipv6 unicast enable
```

```
RP/0/RP0/cpu 0: router(config)#rcc ipv6 unicast period 500
```

または

```
RP/0/RP0/cpu 0: router(config)#lcc ipv6 unicast enable

RP/0/RP0/cpu 0: router(config)#lcc ipv6 unicast period 500
```

RCC または LCC バックグラウンド スキャンをトリガーします。検証のトリガー頻度を制御するには、**period** オプションを使用します。スキャンをトリガーするたびに、転送情報ベース（FIB）に送信されたルートまたはラベルの残りの 1 バッファ分の場所から検証が再開されます。

ステップ 3 **commit**

ステップ 4 次のいずれかのコマンドを使用します。

- **show rcc** {**ipv4**| **ipv6**} **unicast** [**summary** | **scan-id** *scan-id-value*]
- **show lcc** {**ipv4**| **ipv6**} **unicast** [**summary** | **scan-id** *scan-id-value*]

例：

```
RP/0/RP0/cpu 0: router#show rcc ipv6 unicast statistics scan-id 120
```

または

```
RP/0/RP0/cpu 0: router#show lcc ipv6 unicast statistics scan-id 120
```

バックグラウンド スキャンに関する統計情報を表示します。

- **summary**：現在進行中のスキャン ID および以前の少数のスキャンの要約を表示します。
- **scan-id** *scan-id-value*：特定のスキャンに関する詳細情報を表示します。

RCC および LCC のイネーブル化：例

次に、ルート整合性チェッカ（RCC）バックグラウンドスキャンを IPv6 ユニキャスト テーブルのスキャンのバッファ間 500 ミリ秒の時間でイネーブルにする例を示します。

```
rcc ipv6 unicast period 500
```

次に、ラベル整合性チェッカ（LCC）バックグラウンドスキャンを IPv6 ユニキャスト テーブルのスキャンのバッファ間 500 ミリ秒の時間でイネーブルにする例を示します。

```
lcc ipv6 unicast period 500
```

次に、vrf1 のサブネット 10.10.0.0/16 のルート整合性チェッカ（RCC）オンデマンド スキャンを行う例を示します。

```
show rcc ipv4 unicast 10.10.0.0/16 vrf vrf 1
```

次に、ラベル整合性チェッカ（LCC）オンデマンドスキャンを IPv6 プレフィックスのすべてのラベルで実行する例を示します。

```
show lcc ipv6 unicast all
```

RIB の参照

この項では、RIBに関する追加の概念情報について説明します。説明する項目は次のとおりです。

BGP およびその他のプロトコルでの RIB データ構造

RIB は、ボーダー ゲートウェイ プロトコル（BGP）や他のユニキャスト ルーティング プロトコルなどの他のルーティングアプリケーションとは異なるプロセスを使用してデータ構造を維持します。ただし、これらのルーティングプロトコルは、RIBが使用するものと似た内部データ構造を使用し、RIBとしてそのデータ構造を内部的に参照することがあります。たとえば、BGP ルートは BGP RIB（BRIB）に保存されます。RIB プロセスは、BGP によって処理される BRIB に関与しません。

パケットを転送するためにラインカードおよび RP によって使用されるテーブルは、転送情報ベース（FIB）と呼ばれます。RIB プロセスは FIB を構築しません。代わりに、RIB はバルクコンテンツ ダウンローダ（BCDL）プロセスによって、FIBプロセスに最適な、選択されたルートのセットをバルク各ラインカードにダウンロードします。続いて、FIB が構築されます。

RIB アドミニストレーティブ ディスタンス

転送は最長プレフィックス照合に基づいて行われます。10.0.2.1 宛てのパケットを転送する場合、マスク /24 は /16 よりも長い（より具体的である）ため、10.0.2.0/24 は 10.0.0.0/16 よりも優先されます。同じプレフィックスと同じ長さを持つ、異なるプロトコルからのルートは、アドミニストレーティブ ディスタンスに基づいて選択されます。たとえば、Open Shortest Path First（OSPF）プロトコルのアドミニストレーティブ ディスタンスは 110、Intermediate System-to-Intermediate System（IS-IS）プロトコルのアドミニストレーティブ ディスタンスは 115 です。IS-IS および OSPF の両方が RIB に 10.0.1.0/24 をダウンロードすると、OSPF のアドミニストレーティブ ディスタンスの方が小さいため、RIB は OSPF ルートを優先します。同じ長さの複数のルート間で選択するためだけにアドミニストレーティブディスタンスが使用されます。

次の表に、一般的なプロトコルのデフォルトのアドミニストレーティブディスタンスを示します。

表 *1:* デフォルトのアドミニストレーティブ ディスタンス

プロトコル	アドミニストレーティブディスタンスのデフォルト
接続されているルートまたはローカルルート	0
スタティック ルート	1
外部 BGP ルート	20
OSPF ルート	110
IS-IS ルート	115
内部 BGP ルート	200

一部のルーティング プロトコル（たとえば、IS-IS、OSPF、BGP など）のアドミニストレーティブ ディスタンスは変更できます。プロトコルのアドミニストレーティブ ディスタンスを変更する適切な方法については、そのプロトコル固有のマニュアルを参照してください。

（注） すべてではなく一部のルータで、プロトコルのアドミニストレーティブディスタンスを変更すると、ルーティングループなどの予想外の動作が発生することがあります。したがって、これは推奨されません。

RIB 統計情報

RIB は、RIB とクライアントとの間でやり取りされるメッセージ（要求）の統計情報をサポートします。プロトコル クライアントは、メッセージを RIB に送信します（たとえば、ルート追加、ルート削除、ネクスト ホップの登録など）。RIB もメッセージを送信します（たとえば、ルート、アドバタイズメント、ネクスト ホップ通知などの再配布）。これらの統計情報は、どのようなメッセージが送信されたかに関して、また送信されたメッセージ数に関する情報を収集するために使用されます。これらの統計情報には、RIBサーバとそのクライアント間で転送される各種メッセージのカウンタが含まれています。統計情報は、show rib statistics コマンドを使用して表示します。

RIBは、次に挙げるような、クライアントから送信されるすべての要求のカウンタを保持します。

- ルートの動作
- テーブルの登録
- ネクストホップの登録

- 再配布の登録
- 属性の登録
- 同期の完了

RIB は、RIB によって送信されるすべての要求のカウンタも保持します。設定は RIB ネクストホップ ダンプニング機能をディセーブルにします。この結果、クライアントが登録したネクスト ホップが解決された、または解決されなかった場合に RIB はクライアントにすぐに通知します。RIB は、要求の結果に関する情報も保持します。

RIB 隔離

RIB 検疫は、ルーティング プロトコルと RIB 間の相互作用における問題を解決します。問題は、ルートが継続的に挿入され、RIB から取り消される場合に発生する、RIB とルーティング プロトコルの間の持続振動です。問題が解決されるまで、CPU使用率にスパイクが生じます。振動に減衰がない場合、プロトコル プロセスおよび RIB プロセスが CPU を多く使用するため、システムのその他の部分に影響を与え、さらにプロトコルおよびRIBのその他の動作の障害となります。この問題は、RIBにルートの特定の組み合わせが受信されて取り込まれた場合に発生します。この問題は、通常、ネットワークの設定が間違っている場合に発生します。ただし、設定ミスはネットワーク全体であるため、単一のルータの設定時に問題を検出できません。

隔離メカニズムでは相互に再帰的なルートが検出されますが、ここで隔離されるのは相互の再帰が完了した最終ルートです。検疫ルートは、相互の再帰が解消したか確認するために定期的に評価されます。再帰が引き続き存在する場合は、ルートは検疫対象のままとなります。再帰が解消した場合は、ルートは検疫対象から外れます。

次の手順を使用して、ルートを隔離します。

1. RIB は問題がある特定のパスがインストールされている場合に検出します。
2. RIB は、そのパスを取り込んだプロトコルに通知を送信します。
3. プロトコルは問題のルートに関する隔離通知を受信すると、そのルートを「隔離中」とマークします。これが BGP ルートである場合、BGP はそのネイバーにルートへの到達可能性をアドバタイズしません。
4. RIB は、すべての検疫対象パスに対して、安全に取り込む（検疫対象から「使用 OK」状態に移行）ことができるようになったかどうかを定期的にテストします。パスが安全に使用できるようになったことを示す通知がプロトコルに送信されます。

ルートとラベルの整合性チェッカ

ルート整合性チェッカおよびラベル整合性チェッカ（RCC/LCC）はコマンドライン ツールです。これは、コントロール プレーンとデータ プレーン ルート間および IOS XR ソフトウェアのラベル プログラミングの整合性を検証するために使用できます。

運用中ネットワークのルータは、転送情報がコントロール プレーン情報と一致しない状態になった可能性があります。この原因は、ルート プロセッサ（RP）とラインカード（LC）間でのファブリック障害または転送障害、または転送情報ベース（FIB）に関する問題である可能性があります。RCC/LCC を使用すると、結果として生じたコントロール プレーンとデータ プレーン間の不整合を識別して詳細情報を出力できます。この情報は、転送問題とトラフィック損失の原因をさらに調査して診断するために使用できます。

RCC/LCC は、2 つのモードで実行できます。RCC/LCC は、適切なコマンド モードをオンデマンドとして使用して 1 回かぎりのスキャンでトリガーする（オンデマンド スキャン）か、通常のルータ動作中にバックグラウンドで定義した間隔で実行するように設定（バックグラウンド スキャン）できます。RCC は、ルーティング情報ベース（RIB）を転送情報ベース（FIB）と比較します。一方、LCC は、ラベル スイッチング データベース（LSD）を FIB と比較します。不整合が検出されると、RCC/LCC 出力では、特定のルートまたはラベルを識別し、検出された不整合のタイプを識別して、さらなるトラブルシューティングに役立つ追加のデータも提供します。

RCC はルート プロセッサで動作します。FIB は、ライン カード上のエラーについてチェックし、最初の 20 のエラー レポートを RCC に送信します。RCC はすべてのノードからエラー レポートを受信し、それらを要約し（完全一致についてチェックし）、2 つのキュー（ソフトまたはハード）に追加します。各キューのエラー レポート数の制限は 1000 で、キューに優先度はありません。RCC/LCC は、異なるノードからの同じエラー（完全一致）を 1 つのエラーとして記録します。RCC/LCC は、エラーのプレフィックス/ラベル、バージョン番号、タイプなどに基づいてエラーを比較します。

オンデマンド スキャン

オンデマンド スキャンでは、ユーザは、特定のテーブルの特定のプレフィックスまたはテーブル内のすべてのプレフィックス関するコマンドライン インターフェイス全体のスキャンを要求します。スキャンはただちに実行され、結果がすぐに発行されます。LCC は LSD でオンデマンド スキャンを実行するのに対し、RCC は VRF 単位で実行します。

バックグラウンド スキャン

バックグラウンド スキャンでは、ユーザはバックグラウンドで実行されるスキャンを設定します。設定は、定期的なスキャンの間隔で構成されます。このスキャンは、単一または複数のテーブルに設定できます。LCC は LSD でバックグラウンド スキャンを実行するのに対し、RCC はデフォルトの VRF または他の VRF に対し実行します。