



仮想ループバックおよびヌルインターフェイスの設定

このモジュールでは、ループバックおよびヌルインターフェイスの設定について説明します。ループバック インターフェイスとヌルインターフェイスは、仮想インターフェイスと見なされます。

仮想インターフェイスは、ルータ内部の論理パケット スイッチング エンティティです。仮想インターフェイスにはグローバルスコープがありますが、関連付けられた位置はありません。代替として、仮想インターフェイスは名前のあとにグローバルに一意的な数字による ID を持ちます。たとえば、Loopback 0、Loopback 1、Loopback 99999 です。この ID は仮想インターフェイスのタイプごとに固有であるため、Loopback 0 と Null 0 の両方を持つことができ、全体として固有な文字列の名前を形成します。

ループバック インターフェイスとヌルインターフェイスのコントロールプレーンは、アクティブルートスイッチプロセッサ (RSP) 上に存在します。設定およびコントロールプレーンは、スタンバイ RSP 上にミラーリングされ、フェールオーバーが発生した場合には、仮想インターフェイスがそれまでのスタンバイに移り、このスタンバイが新たにアクティブ RSP となります。

- [仮想インターフェイスの設定に関する情報 \(1 ページ\)](#)

仮想インターフェイスの設定に関する情報

仮想インターフェイスを設定するには、次の概念を理解している必要があります。

仮想ループバック インターフェイスの概要

仮想ループバック インターフェイスは、常にアップ状態にあるシングルエンドポイントを持つ仮想インターフェイスです。仮想ループバック インターフェイスで転送されるパケットは、ただちに同じインターフェイスによって受信されます。ループバック インターフェイスは物理インターフェイスをエミュレートします。

Cisco IOS XR ソフトウェアでは、仮想ループバック インターフェイスが次の機能を実行します。

- ループバック インターフェイスは、ルーティング プロトコルセッションの終端アドレスとして設定することができます。これにより、アウトバウンドインターフェイスがダウンしても、ルーティング プロトコルセッションをアップ状態に維持することができます。
- ルータ IP スタックが適切に動作していることを確認するには、ループバック インターフェイスに対して ping を実行します。

他のルータまたはアクセス サーバが仮想ループバック インターフェイスにアクセスを試みるようなアプリケーションでは、ルーティング プロトコルを設定して、ループバック アドレスに割り当てられるサブネットを分散させる必要があります。

ループバック インターフェイスにルーティングされたパケットは、ルータまたはアクセスサーバに再ルーティングされ、ローカルで処理されます。ループバック インターフェイス外にルーティングされるものの、ループバック インターフェイス宛てで送信されない IP パケットは、ドロップされます。これらの2つの状況では、ループバック インターフェイスはヌルインターフェイスのように動作できます。

仮想インターフェイスの設定の前提条件

適切なタスク ID を含むタスク グループに関連付けられているユーザグループに属している必要があります。このコマンドリファレンスには、各コマンドに必要なタスク ID が含まれます。ユーザグループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

仮想ループバック インターフェイスの設定

ここでは、基本的なループバック インターフェイスの設定手順について説明します。

制約事項

ループバック インターフェイスの IP アドレスは、ネットワーク上のすべてのルータ間で固有である必要があります。この IP アドレスは、ルータ上の他のインターフェイスでは使用できません。また、ネットワーク上のいかなるルータのインターフェイスでも使用できません。

手順

ステップ 1 **configure**

例：

```
RP/0/RP0/cpu 0: router# configure
```

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。

ステップ 2 **interface loopback instance**

例：

```
RP/0/RP0/cpu 0: router#(config)# interface Loopback 3
```

インターフェイス コンフィギュレーション モードを開始して、新しいループバック インターフェイスの名前を指定します。

ステップ3 **ipv4 address ip-address**

例：

```
RP/0/RP0/cpu 0: router(config-if)# ipv4 address 100.100.100.69 255.255.255.255
```

ipv4 address コンフィギュレーション コマンドを使用して、仮想ループバック インターフェイスに IP アドレスとサブネット マスクを割り当てます。

ステップ4 **end** または **commit**

例：

```
RP/0/RP0/cpu 0: router(config-if)# end
```

または

```
RP/0/RP0/cpu 0: router(config-if)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されません。

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?  
[cancel]:
```

- **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
- **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
- **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

ステップ5 **show interface type instance**

例：

```
RP/0/RP0/cpu 0: router# show interfaces Loopback0
```

(任意) ループバック インターフェイスの設定を表示します。

例

次に、ループバック インターフェイスを設定する例を示します。

```
RP/0/RP0/cpu 0: router# configure
RP/0/RP0/cpu 0: router(config)# interface Loopback0
RP/0/RP0/cpu 0: router(config-if)# ipv4 address 100.100.100.69 255.255.255.255
RP/0/RP0/cpu 0: router(config-if)# ipv6 address 100::69/128
RP/0/RP0/cpu 0: router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
RP/0/RP0/cpu 0: router# show interfaces Loopback0

Loopback0 is up, line protocol is up
  Interface state transitions: 1
  Hardware is Loopback interface(s)
  Internet address is 100.100.100.69/32
  MTU 1500 bytes, BW 0 Kbit
    reliability Unknown, txload Unknown, rxload Unknown
  Encapsulation Loopback, loopback not set,
  Last link flapped 01:57:47
  Last input Unknown, output Unknown
  Last clearing of "show interface" counters Unknown
  Input/output data rate is disabled.
```

ヌルインターフェイスの概要

ヌルインターフェイスは、ほとんどのオペレーティングシステムで使用可能なヌル装置と同様に機能します。このインターフェイスは常にアップで、トラフィックの転送や受信はできません。カプセル化は常に失敗します。ヌルインターフェイスは、トラフィックをフィルタリングするための代替的な方法として使用できます。不要なネットワークトラフィックをヌルインターフェイスに送ることによって、アクセスリストを使用する場合に伴うオーバーヘッドを回避できます。

ヌルインターフェイスに指定できるインターフェイス コンフィギュレーション コマンドは **ipv4 unreachable** コマンドのみです。 **ipv4 unreachable** コマンドを使用した場合、ソフトウェアは、認識できないプロトコルが使用されている自分宛の非ブロードキャストパケットを受信すると、インターネット制御メッセージプロトコル (ICMP) プロトコル到達不能メッセージを送信元に送信します。宛先アドレスまでのルートが不明なため最終的な宛先に配信できないデータグラムを受信した場合、ソフトウェアはそのデータグラムの発信者に ICMP ホスト到達不能メッセージで応答します。デフォルトでは、 **ipv4 unreachable** コマンドはイネーブルになっています。ICMP にプロトコル到達不能を送信させない場合は、 **ipv4 icmp unreachable disable** コマンドを使用して設定する必要があります。

ブートプロセス時にデフォルトで Null0 インターフェイスが作成されます。このインターフェイスは削除できません。このインターフェイスに **ipv4 unreachable** コマンドを設定することは可能ですが、このインターフェイスは送られてきたすべてのパケットを廃棄するだけなので、ほとんどの設定は不要です。

Null 0 インターフェイスを表示するには、 **show interfaces null0** コマンドを使用します。

ヌルインターフェイスの設定

ここでは、基本的なヌルインターフェイスの設定方法について説明します。

手順

ステップ1 **configure**

例：

```
RP/0/RP0/cpu 0: router# configure
```

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。

ステップ2 **interface null 0**

例：

```
RP/0/RP0/cpu 0: router(config)# interface null 0
```

null0 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。

ステップ3 **end** または **commit**

例：

```
RP/0/RP0/cpu 0: router(config-null0)# end
```

または

```
RP/0/RP0/cpu 0: router(config-null0)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。

```
Uncommitted changes found, commit them before  
exiting(yes/no/cancel)?  
[cancel]:
```

- **yes** と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
- **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
- **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーションファイルに設定変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

ステップ4 show interfaces null 0

例：

```
RP/0/RP0/cpu 0: router# show interfaces null 0
```

ヌルインターフェイスの設定を確認します。

例

次に、ヌルインターフェイスを設定する例を示します。

```
RP/0/RP0/cpu 0: router# configure
RP/0/RP0/cpu 0: router(config)# interface Null 0
RP/0/RP0/cpu 0: router(config-null0)# ipv4 icmp unreachable disable
RP/0/RP0/cpu 0: router(config-null0)# end
Uncommitted changes found, commit them? [yes]: yes
RP/0/RP0/cpu 0: router# show interfaces Null 0
```

```
Null0 is up, line protocol is up
Interface state transitions: 1
Hardware is Null interface
Internet address is Unknown
MTU 1500 bytes, BW 0 Kbit
reliability 255/255, txload Unknown, rxload Unknown
Encapsulation Null, loopback not set,
Last link flapped 4d20h
Last input never, output never
Last clearing of "show interface" counters 05:42:04
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 total input drops
0 drops for unrecognized upper-level protocol
Received 0 broadcast packets, 0 multicast packets
0 packets output, 0 bytes, 0 total output drops
Output 0 broadcast packets, 0 multicast packets
```

仮想 IPv4 インターフェイスの設定

ここでは、IPv4 仮想インターフェイスの設定手順について説明します。

手順

ステップ1 configure

例：

```
RP/0/RP0/cpu 0: router# configure
```

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。

ステップ2 ipv4 virtual address ipv4-

例：

```
RP/0/RP0/cpu 0: router(config)# ipv4 virtual address 10.3.32.154/8
```

管理イーサネットインターフェイスの IPv4 仮想アドレスを定義します。

ステップ 3 end または commit

例：

```
RP/0/RP0/cpu 0: router(config-null0)# end
```

または

```
RP/0/RP0/cpu 0: router(config-null0)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。

```
Uncommitted changes found, commit them before  
exiting(yes/no/cancel)?  
[cancel]:
```

- **yes** と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
- **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
- **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーションファイルに設定変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

例

次に、仮想 IPv4 インターフェイスを設定する例を示します。

```
RP/0/RP0/cpu 0: router# configure  
RP/0/RP0/cpu 0: router(config)# ipv4 virtual address 10.3.32.154/8  
RP/0/RP0/cpu 0: router(config-null0)# commit
```

