IS-IS の実装

Integrated Intermediate System-to-Intermediate System（IS-IS）、インターネット プロトコル バージョン 4（IPv4）は、標準ベースの内部ゲートウェイ プロトコル（IGP）です。Cisco ソフトウェアは、国際標準化機構（ISO）/International Engineering Consortium（IEC）10589 および RFC 1195 に記載されている IP ルーティング機能を実装し、IP バージョン 6（IPv6）向けに標準拡張のシングル トポロジおよびマルチトポロジ IS-IS を追加しています。

このモジュールでは、Cisco IOS XR ネットワークで IS-IS（IPv4 および IPv6）を実装する方法について説明します。

IS-IS のイネーブル化およびレベル 1 またはレベル 2 ルーティングの設定

ここでは、IS-IS をイネーブルにし、エリアのルーティング レベルを設定する方法について説明します。

（注） ステップ 4 のルーティング レベルの設定は任意ですが、適切なレベルの隣接関係を確立するために設定することを強く推奨します。

始める前に

IP アドレスを設定する前に IS-IS を設定できますが、少なくとも 1 つの IP アドレスを設定するまでは IS-IS ルーティングは行われません。

手順

ステップ 1 **configure**

ステップ 2 **router isis** *instance-id*

例：

```
RP/0/RP0/CPU0:router(config)# router isis isp
```

指定したルーティング インスタンスの IS-IS ルーティングをイネーブルにし、ルータをルータ コンフィギュレーション モードにします。

- 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 とレベル 2 になります。**is-type** ルータ コンフィギュレーション コマンドを使用して、特定のルーティング インスタンスによって実行されるルーティングのレベルを変更できます。

ステップ 3 **net** *network-entity-title*

例：

```
RP/0/RP0/CPU0:router(config-isis)# net 47.0004.004d.0001.0001.0c11.1110.00
```

ルーティング インスタンスの Network Entity Title（NET）を設定します。

- マルチインスタンスの IS-IS を設定する場合は、ルーティング インスタンスごとに NET を指定します。
- この例では、エリア ID が 47.0004.004d.0001 でシステム ID が 0001.0c11.1110.00 のルータを設定します。
- 複数のエリア アドレスを指定するには、追加の NET を指定します。NET のエリア アドレス部分が異なる場合でも、システム ID 部分はすべての設定項目で完全に一致する必要があります。

ステップ 4 **is-type** { **level-1** | **level-1-2** | **level-2-only** }

例：

```
RP/0/RP0/CPU0:router(config-isis)# is-type level-2-only
```

（任意）システム タイプ（エリアまたはバックボーン ルータ）を設定します。

- デフォルトでは、すべての IS-IS インスタンスは **level-1-2** ルータとして動作します。
- **level-1** キーワードは、レベル 1（エリア内）ルーティングのみを実行するようにソフトウェアを設定します。レベル 1 の隣接関係のみが確立されます。ソフトウェアはエリア内の宛先についてのみ学習します。エリア外の宛先を含むすべてのパケットは、エリア内の最も近い **level-1-2** ルータに送信されます。
- **level-2-only** キーワードは、レベル 2（バックボーン）ルーティングのみを実行するようにソフトウェアを設定します。ルータは、他のレベル 2 のみのルータまたは **レベル 1 と 2** のルータとの間でレベル 2 の隣接関係のみを確立します。
- **level-1-2** キーワードは、レベル 1 とレベル 2 の両方のルーティングを実行するようにソフトウェアを設定します。レベル 1 とレベル 2 の両方の隣接関係が確立されます。ルータはレベル 2 バックボーンとレベル 1 エリアの間の境界ルータとして動作します。

ステップ 5 **commit**

ステップ 6 **show isis** [**instance** *instance-id*] **protocol**

例：

```
RP/0/RP0/CPU0:router# show isis protocol
```

（任意）IS-IS インスタンスに関するサマリー情報を表示します。

単一トポロジ IPv6

単一トポロジ IPv6 により、インターフェイス上で IPv4 ネットワーク プロトコルに加えて IPv6 用の IS-IS を設定できます。すべてのインターフェイスは同一のネットワーク プロトコル セットで構成されている必要があります。また、IS-IS エリア（レベル 1 ルーティング用）またはドメイン（レベル 2 ルーティング用）のすべてのルータは、すべてのインターフェイスで同一のネットワーク層プロトコル セットをサポートする必要があります。

single-topology モードでは、IPv6 トポロジは IPv4 ユニキャスト トポロジのナロー、ワイド メトリック スタイルの両方で機能します。single-topology での動作中は、レベルごとに 1 つの Shortest Path First（SPF）の計算が IPv4 ルートと IPv6 ルートの両方の計算に使用されます。IPv4 IS-IS と IPv6 IS-IS のルーティング プロトコルが共通のリンク トポロジを共有するため、単一の SPF の使用が可能です。

IS-IS のシングル トポロジの設定

IS-IS インスタンスをイネーブルにした後で、特定のネットワーク トポロジのルートを計算するように設定する必要があります。

ここでは、IPv4 または IPv6 トポロジ向けのインターフェイスで IS-IS プロトコルの動作を設定する方法について説明します。

始める前に

（注） ルータを single-topology モードで実行できるようにするには、IS-IS の各インターフェイスですべてのアドレス ファミリをイネーブルに設定し、IS-IS ルータ スタンザの IPv6 ユニキャスト アドレスファミリ内で「single-topology」を設定します。IPv6 アドレス ファミリ、または IPv4 と IPv6 の両方のアドレス ファミリを使用できますが、設定ではルータ上のすべてのアクティブなアドレス ファミリ セットを表します。さらに IPv6 ルータ アドレス ファミリ サブモードで single-topology を設定して、明示的に single-topology 動作をイネーブルにします。

これらの手順には例外が 2 つあります。

1. IS-IS プロセスの address-family スタンザに adjacency-check disable コマンドが含まれる場合、インターフェイスでアドレス ファミリをイネーブルにする必要はありません。
2. single-topology コマンドは ipv4 アドレス ファミリ サブモードでは無効です。

シングル トポロジのデフォルトのメトリック スタイルはナロー メトリックです。ワイド メトリックまたはナロー メトリックのどちらかを使用できます。この設定方法はシングル トポロジの設定に依存します。IPv4 と IPv6 の両方がイネーブルでシングル トポロジが設定されている場合には、メトリック スタイルは **address-family ipv4** スタンザ内で設定します。メトリックは **address-family ipv6** スタンザ内でも設定できますが、この場合には設定は無視されます。IPv6 のみがイネーブルでシングル トポロジが設定されている場合には、メトリック スタイルは **address-family ipv6** スタンザ内で設定します。

手順

ステップ 1 **configure**

ステップ 2 **interface** *type interface-path-id*

例：

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE 0/9/0/0
```

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 次のいずれかを実行します。

- **ipv4 address** *address mask*
- **ipv6 address** *ipv6-prefix* / *prefix-length* [**eui-64**]
- **ipv6 address** *ipv6-address* { / *prefix-length* | *link-local* }
- **ipv6 enable**

例：

```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.0.1.3 255.255.255.0
```

または

```
RP/0/RP0/CPU0:router(config-if)# ipv6 address 3ffe:1234:c18:1::/64 eui-64
RP/0/RP0/CPU0:router(config-if)# ipv6 address FE80::260:3EFF:FE11:6770 link-local
RP/0/RP0/CPU0:router(config-if)# ipv6 enable
```

または

インターフェイスの IPv4 アドレスを定義します。インターフェイスのいずれかで IS-IS ルーティングが設定されている場合は、IS-IS がイネーブルになっているエリアに含まれるすべてのインターフェイスで IP アドレスが必要です。

または

インターフェイスに割り当てられた IPv6 ネットワークを指定し、**eui-64** キーワードでインターフェイスの IPv6 処理をイネーブルにします。

または

インターフェイスに割り当てられた IPv6 インターフェイスを指定し、**link-local** キーワードでインターフェイスの IPv6 処理をイネーブルにします。

または

インターフェイスで IPv6 リンクローカル アドレスを自動的に設定し、インターフェイスで IPv6 処理もイネーブルにします。

- リンクローカル アドレスは、同じリンク上のノードとの通信にだけ使用できます。
- ipv6 address ipv6-prefix / prefix-length インターフェイス コンフィギュレーション コマンドを **eui-64** キーワードを付けずに指定すると、サイトローカルのグローバル IPv6 アドレスが設定されます。
- ipv6 address ipv6-prefix / prefix-length コマンドを **eui-64** キーワードとともに指定すると、IPv6 アドレスの下位 64 ビットにインターフェイス ID を持つサイトローカルのグローバル 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 コマンドを **link-local** キーワードとともに指定すると、IPv6 がインターフェイスでイネーブルになっている場合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るリンクローカル アドレスがインターフェイスに設定されます。

ステップ 4 **exit**

例：

```
RP/0/RP0/CPU0:router(config-if)# exit
```

インターフェイス コンフィギュレーション モードを終了し、ルータを モードに戻します。

ステップ 5 **router isis** *instance-id*

例：

```
RP/0/RP0/CPU0:router(config)# router isis isp
```

指定したルーティング インスタンスの IS-IS ルーティングをイネーブルにし、ルータをルータ コンフィギュレーション モードにします。

- デフォルトでは、すべての IS-IS インスタンスがレベル 1 とレベル 2 になります。**is-type** コマンドを使用して、特定のルーティング インスタンスによって実行されるルーティングのレベルを変更できます。

ステップ 6 **net** *network-entity-title*

例：

```
RP/0/RP0/CPU0:router(config-isis)# net 47.0004.004d.0001.0001.0c11.1110.00
```

ルーティング インスタンスの NET を設定します。

- マルチインスタンスの IS-IS を設定する場合は、ルーティング インスタンスごとに NET を指定します。NET およびアドレスの名前を指定できます。
- この例では、エリア ID が 47.0004.004d.0001 でシステム ID が 0001.0c11.1110.00 のルータを設定します。
- 複数のエリア アドレスを指定するには、追加の NET を指定します。NET のエリア アドレス部分が異なる場合でも、システム ID 部分はすべての設定項目で完全に一致する必要があります。

ステップ 7 **address-family ipv6** [**unicast**]

例：

```
RP/0/RP0/CPU0:router(config-isis)# address-family ipv6 unicast
```

IPv6 アドレス ファミリを指定し、ルータ アドレス ファミリ コンフィギュレーション モードを開始します。

- この例では、ユニキャスト IPv6 アドレス ファミリを指定します。

ステップ 8 **single-topology**

例：

```
RP/0/RP0/CPU0:router(config-isis-af)# single-topology
```

（任意）IPv6 が設定されているときに IPv4 のリンク トポロジを設定します。

- **single-topology** コマンドは IPv6 サブモードでのみ有効です。このコマンドは、マルチトポロジ モードでデフォルトの設定である分離されたトポロジではなく、シングル トポロジを使用することを IPv6 に指示します。

ステップ 9 exit

例：

```
RP/0/RP0/CPU0:router(config-isis-af)# exit
```

ルータ アドレス ファミリ コンフィギュレーション モードを終了して、ルータをルータ コンフィギュレーション モードに戻します。

ステップ 10 **interface** *type interface-path-id*

例：

```
RP/0/RP0/CPU0:router(config-isis)# interface HundredGigE 0/9/0/0
```

インターフェイス設定モードを開始します。

ステップ 11 **circuit-type** { **level-1** | **level-1-2** | **level-2-only** }

例：

```
RP/0/RP0/CPU0:router(config-isis-if)# circuit-type level-1-2
```

（任意）隣接関係のタイプを設定します。

- デフォルトの回路タイプは設定済みの（**is-type** コマンドで設定した）システム タイプです。
- 通常、ルータを **level-1-2** のみとして設定し、**level-1** または **level-2-only** のみの隣接関係を形成するようにインターフェイスを制約する場合は、回線タイプを設定する必要があります。

ステップ 12 **address-family** { **ipv4** | **ipv6** } [**unicast**]

例：

```
RP/0/RP0/CPU0:router(config-isis-if)# address-family ipv4 unicast
```

IPv4 または IPv6 アドレス ファミリを指定して、インターフェイス 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。

- この例では、インターフェイスにユニキャスト IPv4 アドレス ファミリを指定します。

ステップ 13 **commit**

ステップ 14 **show isis** [**instance** *instance-id*] **interface** [*type interface-path-id*] [**detail**] [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router# show isis interface HundredGigE 0/9/0/0
```

（任意）IS-IS インターフェイスに関する情報を表示します。

ステップ 15 **show isis** [**instance** *instance-id*] **topology** [**systemid** *system-id*] [**level** { **1** | **2** }] [**summary**]

例：

```
RP/0/RP0/CPU0:router# show isis topology
```

（任意）すべてのエリアの接続済みルータのリストを表示します。

シングルトポロジ IS-IS for IPv6 の設定：例

次に、single-topology モードのイネーブル化の例を示します。IS-IS インスタンスが作成され、NET が定義され、インターフェイス上で IPv6 が IPv4 とともに設定され、IPv4 リンク トポロジが IPv6 で使用されます。この設定は、POS インターフェイス 0/3/0/0 が IPv4 アドレスと IPv6 アドレスの両方の隣接関係を形成できるようにします。

```
router isis isp
 net 49.0000.0000.0001.00
 address-family ipv6 unicast
  single-topology
 interface tenGigE 0/11/0/0
  address-family ipv4 unicast
  !
  address-family ipv6 unicast
  !
  exit
!
interface tenGigE 0/11/0/0
 ipv4 address 10.0.1.3 255.255.255.0
 ipv6 address 2001::1/64
```

単一トポロジ構成用の SPF 間隔の設定

ここでは、ルータのパフォーマンスをチューニングするために SPF 計算を調整する方法について説明します。このタスクはオプションです。

SPF 計算は特定のトポロジのルートを計算するため、チューニング属性はルータ アドレス ファミリ コンフィギュレーション サブモード内にあります。SPF 計算は、レベル 1 とレベル 2 のルートを別個に計算します。

IPv4 と IPv6 のアドレス ファミリが single-topology モードで使用される場合には、IPv4 トポロジ用の 1 つの SPF だけが存在します。IPv6 トポロジは IPv4 のトポロジを「借用」するため、IPv6 用の SPF 計算は必要ありません。single-topology モードで SPF 計算のパラメータを調整するには、address-family ipv4 unicast コマンドを設定します。

Incremental SPF アルゴリズムは、個別にイネーブルにできます。Incremental Shortest Path First（ISPF）は、イネーブルにしたときにすぐには適用されません。代わりにフル SPF アルゴリズムが使用されて、ISPF の実行に必要な状態情報の「シード」が作成されます。起動遅延により、IS-IS 再起動後の ISPF の実行が指定された期間止められます（データベースを安定させるため）。起動遅延期間が経過した後は、ISPF がすべての SPF 計算の実行について主要な役割を担います。シード更新間隔は、iSFP の状態の同期を維持するためにフル SPF の定期的な実行を可能にします。

手順

ステップ 1 **configure**

ステップ 2 **router isis** *instance-id*

例：

```
RP/0/RP0/CPU0:router(config)# router isis isp
```

指定したルーティング インスタンスの IS-IS ルーティングをイネーブルにし、ルータをルータ コンフィギュレーション モードにします。

- **is-type** ルータ コンフィギュレーション コマンドを使用して、特定のルーティング インスタンスによって実行されるルーティングのレベルを変更できます。

ステップ 3 **address-family** { **ipv4** | **ipv6** } [**unicast**]

例：

```
RP/0/RP0/CPU0:router(config-isis)#address-family ipv4 unicast
```

IPv4 または IPv6 アドレス ファミリを指定して、ルータ アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 4 **spf-interval** {[**initial-wait** *initial* | **secondary-wait** *secondary* | **maximum-wait** *maximum*] ...} [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis-af)# spf-interval initial-wait 10 maximum-wait 30
```

（任意）連続する SPF 計算の最小間隔を制御します。

- この値は、イベントがトリガーされた後の SPF 計算を遅延させ、SPF の実行の間に最小経過時間を適用させます。
- 小さすぎる値が設定された場合には、ネットワークが不安定なときにルータが大量の CPU リソースを失う可能性があります。
- 大きすぎる値が設定された場合には、ネットワーク トポロジの変更が遅延し、パケットを損失する結果になります。
- ISPF アルゴリズムは変更された LSP を受信するたびにすぐ実行されるため、SPF 間隔は ISPF の実行には適用されません。

ステップ 5 **ispf** [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis-af)# ispf
```

（任意）Incremental IS-IS ISPF がネットワーク トポロジを計算するように設定します。

ステップ 6 **commit**

ステップ 7 **show isis** [**instance** *instance-id*] [[**ipv4** | **ipv6** | **afi-all**] [**unicast** | **safi-all**]] **spf-log** [**level** { **1** | **2** }] [**ispf** | **fspf** | **prc** | **nhc**] [**detail** | **verbose**] [**last** *number* | **first** *number*]

例：

```
RP/0/RP0/CPU0:router# show isis instance 1 spf-log ipv4
```

（任意）ルータがフル SPF 計算を実行した頻度と、実行理由を表示します。

IS-IS のルートのカスタマイズ

ここでは、ルート機能を実行する方法について説明します。デフォルト ルートを IS-IS ルーティング ドメインに挿入する機能や別の IS-IS インスタンスで学習されたルートを再配布する機能が含まれます。このタスクはオプションです。

手順

ステップ 1 **configure**

ステップ 2 **router isis** *instance-id*

例：

```
RP/0/RP0/CPU0:router(config)# router isis isp
```

指定したルーティング プロセスの IS-IS ルーティングをイネーブルにし、ルータをルータ コンフィギュレーション モードにします。

- 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 とレベル 2 になります。**is-type** コマンドを使用して、特定のルーティング インスタンスによって実行されるルーティングのレベルを変更できます。

ステップ 3 **set-overload-bit** [**on-startup** { *delay* | **wait-for-bgp** }] [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis)# set-overload-bit
```

（任意）過負荷ビットを設定します。

（注） NSF 再起動が再起動中に過負荷ビットを設定しないため、設定された過負荷ビットの動作は NSF の再起動に適用されません。

ステップ 4 **address-family** { **ipv4** | **ipv6** } [**unicast**]

例：

```
RP/0/RP0/CPU0:router(config-isis)# address-family ipv4 unicast
```

IPv4 または IPv6 アドレス ファミリを指定して、ルータ アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 5 **default-information originate** [**route-policy** *route-policy-name*]

例：

```
RP/0/RP0/CPU0:router(config-isis-af)# default-information originate
```

（任意）IPv4 または IPv6 のデフォルト ルートを IS-IS ルーティング ドメインに挿入します。

- **route-policy** キーワードと *route-policy-name* 引数により、IPv4 または IPv6 のデフォルト ルートをアドバタイズする条件を指定します。
- **route-policy** キーワードを省略すると、IPv4 または IPv6 のデフォルト ルートは無条件にレベル 2 でアドバタイズされます。

ステップ 6 **distribute-list** { {**prefix-list** *prefix-list-name* | **route-policy** *route-policy-name*} } **in**

例：

```
RP/0/RP0/CPU0:router(config-isis)# distribute-list { {prefix-list | prefix-list-name} |
 {route-policy | route-policy-name} } in
```

（任意）Intermediate System-to-Intermediate System（IS-IS）がルーティング情報ベース（RIB）にインストールするルートをフィルタリングします。

警告 **distribute-list in** コマンドが設定されている場合、IS-IS で計算される一部のルートはローカル ルータのフォワーディング プレーンにインストールされませんが、他の IS-IS ルータはこれを認識しません。このため、他の IS-IS ルータで計算されたフォワーディング ステートとこのルータ上の実際のフォワーディング ステートに違いが生まれます。場合によっては、トラフィックがドロップまたはループする可能性があります。このため、このコマンドを使用するタイミングに注意してください。

ステップ 7 **redistribute isis** *instance* [**level-1** | **level-2** | **level-1-2**] [**metric** *metric*] [**metric-type** { **internal** | **external** }] [**policy** *policy-name*]

例：

```
RP/0/RP0/CPU0:router(config-isis-af)# redistribute isis 2 level-1
```

（任意）ある IS-IS インスタンスから別のインスタンスにルートを再配布します。

- この例では、IS-IS インスタンスは別の IS-IS インスタンスからのレベル 1 ルートを再配布します。

ステップ 8 次のいずれかを実行します。

- **summary-prefix** *address* / *prefix-length* [**level** { **1** | **2** }]
- **summary-prefix** *ipv6-prefix* / *prefix-length* [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis-af)# summary-prefix 10.1.0.0/16 level 1
```

または

```
RP/0/RP0/CPU0:router(config-isis-af)# summary-prefix 3003:xxxx::/24 level 1
```

（任意）レベル 1-2 ルータがサマリーをアドバタイズするときに直接レベル 1 プレフィックスをアドバタイズするのではなく、レベル 1 IPv4 および IPv6 プレフィックスをレベル 2 で集約できるようにします。

- この例では、IPv4 アドレスおよびマスクを指定します。

または

- この例では IPv6 プレフィックスを指定し、コマンドは RFC 2373 に記載された形式にする必要があり、16 ビット値をコロンで区切った 16 進でアドレスを指定します。
- IPv6 プレフィックスは IPv6 ルータ アドレス ファミリ コンフィギュレーション サブモードでのみ設定でき、IPv4 プレフィックスは IPv4 ルータ アドレス ファミリ コンフィギュレーション サブモードでのみ設定できます。

ステップ 9 **maximum-paths** *route-number*

例：

```
RP/0/RP0/CPU0:router(config-isis-af)# maximum-paths 16
```

（任意）ルーティング テーブルで許可されるパラレル パスの最大数を設定します。

ステップ 10 **distance** *weight* [*address / prefix-length* [*route-list-name*]]

例：

```
RP/0/RP0/CPU0:router(config-isis-af)# distance 90
```

（任意）IS-IS プロトコルにより発見されたルートに割り当てられるアドミニストレーティブ ディスタンスを定義します。

- IPv4 と IPv6 で異なるアドミニストレーティブ ディスタンスを適用できます。

ステップ 11 **set-attached-bit**

例：

```
RP/0/RP0/CPU0:router(config-isis-af)# set-attached-bit
```

（任意）IS-IS インスタンスにレベル 1 LSP 内の attached ビットを設定します。

ステップ 12 **commit**

複数インスタンス間での IS-IS ルートの再配布：例

次に、set-attached-bit および redistribute コマンドの使用例を示します。レベル 1 に制限されたインスタンス「1」とレベル 2 に制限されたインスタンス「2」の 2 つのインスタンスが設定されています。

再配布を使用してレベル 1 のインスタンスからレベル 2 のインスタンスにルートが伝播します。レベル 1 のルートが優先されるように、レベル 2 インスタンスのアドミニストレーティブ ディスタンスが明示的に大きく設定されていることに注目してください。

レベル 1 インスタンスはレベル 2 インスタンスへの再配布ルートであることから、レベル 1 インスタンスには attached ビットが設定されています。このため、インスタンス「1」はエリアからバックボーンへ到達するための適切な候補になります。

```
  router isis 1
    is-type level-2-only
   net 49.0001.0001.0001.0001.00
   address-family ipv4 unicast
    distance 116
    redistribute isis 2 level 2
  !
  interface HundredGigE 0/9/0/0
   address-family ipv4 unicast
  !
  !
  router isis 2
   is-type level-1
   net 49.0002.0001.0001.0002.00
   address-family ipv4 unicast
    set
-attached-bit

  !
  interface HundredGigE 0/9/0/0
   address-family ipv4 unicast
```

RIB にプレフィックスを追加するためのプライオリティの設定

このオプションの手順では、指定されたプレフィックスを RIB に追加するプライオリティ（順序）の設定方法について説明します。プレフィックスは、アクセスリスト（ACL）、プレフィックス リスト、またはタグ値の照合を使用して選択できます。

手順

ステップ 1 **configure**

ステップ 2 **router isis** *instance-id*

例：

```
RP/0/RP0/CPU0:router(config)# router isis isp
```

指定したルーティング プロセスの IS-IS ルーティングをイネーブルにし、ルータをルータ コンフィギュレーション モードにします。この例では、IS-IS インスタンスは isp と呼ばれます。

ステップ 3 **address-family** { **ipv4** | **ipv6** } [**unicast**]

例：

```
RP/0/RP0/CPU0:router(config-isis)# address-family ipv4 unicast
```

IPv4 または IPv6 アドレス ファミリを指定して、ルータ アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 4 **metric-style wide** [**transition**] [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis-af)# metric-style wide level 1
```

レベル 1 エリアでワイド リンク メトリックのみを生成して受け入れるようにルータを設定します。

ステップ 5 **spf prefix-priority** [**level** { **1** | **2** }] { **critical** | **high** | **medium** } { *access-list-name* | **tag** *tag* }

例：

```
RP/0/RP0/CPU0:router(config-isis-af)# spf prefix-priority high tag 3
```

値が 3 のタグが付けられたすべてのルートを先にインストールします。

ステップ 6 **commit**

IS-IS のインターフェイス

IS-IS のインターフェイスは次のタイプのいずれかとして設定できます。

- アクティブ：接続されたプレフィックスをアドバタイズし、隣接関係を形成します。これはデフォルトのインターフェイスです。
- パッシブ：接続されたプレフィックスをアドバタイズしますが、隣接関係は形成しません。インターフェイスをパッシブに設定するには、passive コマンドを使用します。パッシブなインターフェイスは、IS-IS ドメインへの挿入が必要なループバック アドレスのような、重要なプレフィックスのために控えめに使用します。多くの接続されたプレフィックスをアドバタイズする必要がある場合には、適切なポリシーを備えた接続ルートの再配布を代わりに使用します。

- 抑制：接続されたプレフィックスをアドバタイズせず、隣接関係を形成します。インターフェイスを抑制に設定するには、suppress コマンドを使用します。
- シャットダウン：接続されたプレフィックスをアドバタイズせず、隣接関係も形成しません。IS-IS の設定を削除せずにインターフェイスをディセーブルにするには、shutdown コマンドを使用します。

IS-IS インターフェイス ルートのタギング

このオプションの手順では、IS-IS インターフェイスの接続されたルートにタグを関連付ける方法について説明します。

手順

ステップ 1 **configure**

ステップ 2 **router isis** *instance-id*

例：

```
RP/0/RP0/CPU0:router(config)# router isis isp
```

指定したルーティング プロセスの IS-IS ルーティングをイネーブルにし、ルータをルータ コンフィギュレーション モードにします。この例では、IS-IS インスタンスは isp と呼ばれます。

ステップ 3 **address-family** { **ipv4** | **ipv6** } [**unicast**]

例：

```
RP/0/RP0/CPU0:router(config-isis)# address-family ipv4 unicast
```

IPv4 または IPv6 アドレス ファミリを指定して、ルータ アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 4 **metric-style wide** [**transition**] [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis-af)# metric-style wide level 1
```

レベル 1 エリアでワイド リンク メトリックのみを生成して受け入れるようにルータを設定します。

ステップ 5 **exit**

例：

```
RP/0/RP0/CPU0:router(config-isis-af)# exit
```

ルータ アドレス ファミリ コンフィギュレーション モードを終了して、ルータをルータ コンフィギュレーション モードに戻します。

ステップ 6 **interface** *type number*

例：

```
RP/0/RP0/CPU0:router(config-isis)# interface HundredGigE 0/9/0/0
```

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 7 **address-family** { **ipv4** | **ipv6** } [**unicast**]

例：

```
RP/0/RP0/CPU0:router(config-isis-if)# address-family ipv4 unicast
```

IPv4 または IPv6 アドレス ファミリを指定して、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。

ステップ 8 **tag** *tag*

例：

```
RP/0/RP0/CPU0:router(config-isis-if-af)# tag 3
```

アドバタイズされた接続されたルートに関連付けるタグの値を設定します。

ステップ 9 **commit**

ステップ 10 **show isis** [**ipv4** | **ipv6** | **afi-all**] [**unicast** | **safi-all**] **route** [**detail**]

例：

```
RP/0/RP0/CPU0:router(config-isis-if-af)# show isis ipv4 route detail
```

タグ情報を表示します。すべてのタグが RIB に存在することを確認します。

ルートのタギング：例

次に、ルートのタギングの例を示します。

```
route-policy isis-tag-55
end-policy
!
route-policy isis-tag-555
  if destination in (5.5.5.0/24 eq 24) then
    set tag 555
    pass
  else
    drop
  endif
end-policy
!
router static
 address-family ipv4 unicast
  0.0.0.0/0 2.6.0.1
  5.5.5.0/24 Null0
 !
```

```
!
router isis uut
 net 00.0000.0000.12a5.00
 address-family ipv4 unicast
  metric-style wide
  redistribute static level-1 route-policy isis-tag-555
  spf prefix-priority critical tag 13
  spf prefix-priority high tag 444
  spf prefix-priority medium tag 777
```

LSP フラッディングの制限

リンク ステート パケット（LSP）を制限すると、特定の「メッシュの」ネットワーク トポロジで有効な場合があります。このようなネットワークの例は、非ブロードキャストマルチアクセス（NBMA）トランスポート上の完全メッシュ化されたポイントツーポイントリンクのセットなどの冗長性の高いネットワークです。このようなネットワークでは、完全な LSP フラッディングにより、ネットワークのスケーラビリティを制限できます。フラッディングのドメインのサイズを制限する 1 つの方法は、複数のレベル 1 エリアと 1 つのレベル 2 エリアを使用することにより、階層を導入することです。ただし、階層の代わりに他の 2 つの技法を使用することもできます。特定のインターフェイス上でフラッディングをブロックし、メッシュ グループを設定します。

両方の技法は、LSP フラッディングを何らかの方法で制限することで動作します。直接的な結果として、ネットワークのスケーラビリティが改善される一方で、ネットワークの（障害時の）信頼性が低下します。ブロッキングやメッシュ グループによって使用が制限されていない場合、フラッディングが可能なリンクが存在しても、一連の障害によって LSP をネットワーク全体にフラッディングできないことがあるからです。このような場合、ネットワーク内の異なるルータのリンク ステート データベースを、同期できないことがあります。永続的な転送ループのような問題が結果として発生する可能性があります。したがって、ブロッキングやメッシュ グループはどうしても必要な場合にかぎり、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。

IS-IS の LSP フラッディングの制御

LSP フラッディングにより、ネットワークのスケーラビリティを制限できます。ルータでグローバルに、またはインターフェイスで LSP データベース パラメータを調整することによって、LSP フラッディングを制御できます。このタスクはオプションです。

LSP フラッディングを制御するコマンドの多くには、適用されるレベルを指定するオプションが含まれます。オプションを指定しなかった場合、コマンドは両方のレベルに適用されます。オプションが 1 つのレベルに設定された場合、もう一方のレベルはデフォルト値を使用し続けます。両方のレベルのオプションを設定するには、コマンド twice を使用します。次に例を示します。

```
RP/0/RP0/CPU0:router(config-isis)# lsp-refresh-interval 1200 level 2
RP/0/RP0/CPU0:router(config-isis)# lsp-refresh-interval 1100 level 1
```

手順

ステップ 1 **configure**

ステップ 2 **router isis** *instance-id*

例：

```
RP/0/RP0/CPU0:router(config)# router isis isp
```

指定したルーティング インスタンスの IS-IS ルーティングをイネーブルにし、ルータをルータ コンフィギュレーション モードにします。

- **is-type** ルータ コンフィギュレーション コマンドを使用して、特定のルーティング インスタンスによって実行されるルーティングのレベルを変更できます。

ステップ 3 **lsp-refresh-interval** *seconds* [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis)# lsp-refresh-interval 10800
```

（任意）異なるシーケンス番号を持つ LSP を再生成する間隔を設定します。

- 更新間隔は、常に、max-lsp-lifetime コマンドよりも低く設定する必要があります。

ステップ 4 **lsp-check-interval** *seconds* [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis)# lsp-check-interval 240
```

（任意）データベースの LSP のチェックサムを検証するデータベース全体の定期チェックの間隔を設定します。

- この操作は、CPU の点でコスト高であるため、あまり発生しないように設定する必要があります。

ステップ 5 **lsp-gen-interval** { [**initial-wait** *initial* | **secondary-wait** *secondary* | **maximum-wait** *maximum*] **...** } [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis)# lsp-gen-interval maximum-wait 15 initial-wait 5
```

（任意）ネットワークが不安定な間は LSP の生成レートを低下させます。ルータの CPU 負荷を軽減し、IS-IS ネイバーへの LSP 送信数を低減するのに役立ちます。

- ネットワークの不安定性が長引いている間に LSP の再計算を繰り返すと、ローカル ルータの CPU 負荷が増加する可能性があります。さらに、これらの再計算された LSP をネッ

トワーク内の他の中継システムにフラッディングすると、トラフィックが増加し、他のルータがルート計算を実行するために費やす時間が増加する可能性があります。

ステップ 6 **lsp-mtu** *bytes* [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis)# lsp-mtu 1300
```

（任意）LSP の最大伝送単位（MTU）サイズを設定します。

ステップ 7 **max-lsp-lifetime** *seconds* [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis)# max-lsp-lifetime 11000
```

（任意）ルータから発信された LSP に設定する最初のライフタイムを設定します。

- これは、LSPが再生成または更新されない場合に、ネイバーのデータベースにLSPが維持される時間です。

ステップ 8 **ignore-lsp-errors disable**

例：

```
RP/0/RP0/CPU0:router(config-isis)# ignore-lsp-errors disable
```

（任意）チェックサム エラーで受信した LSP をパージするようにルータを設定します。

ステップ 9 **interface** *type interface-path-id*

例：

```
RP/0/RP0/CPU0:router(config-isis)# interface HundredGigE 0/9/0/0
```

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 10 **lsp-interval** *milliseconds* [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis-if)# lsp-interval 100
```

（任意）インターフェイス上で送信された各 LSP 間の時間を設定します。

ステップ 11 **csnp-interval** *seconds* [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis-if)# csnp-interval 30 level 1
```

（任意）ブロードキャストインターフェイス上で定期的にCSNPパケットが送信される間隔を設定します。

- より頻繁にCSNPを送信することは、受信のために隣接ルータはより激しく動作する必要があることを意味します。
- CSNP の送信の頻度を下げることは、隣接ルータ間の相違がより長く続くことを意味します。

ステップ 12 **retransmit-interval** *seconds* [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis-if)# retransmit-interval 60
```

（任意）LSPが受信されていないと判断して再送信するまでに送信ルータが応答を待つ時間を設定します。

```
RP/0/RP0/CPU0:router(config-isis-if)# retransmit-interval 60
```

ステップ 13 **retransmit-throttle-interval** *milliseconds* [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis-if)# retransmit-throttle-interval 1000
```

（任意）ポイントツーポイント インターフェイス上の各 LSP の再送信間隔を設定します。

- この時間は通常 lsp-interval コマンドの時間以上にします。これは隣接ルータがビジーであることがLSPが失われた原因の可能性があるためです。間隔を長くするとネイバーはより時間をかけて送信を受け取ることができます。

ステップ 14 **mesh-group** { *number* | **blocked** }

例：

```
RP/0/RP0/CPU0:router(config-isis-if)# mesh-group blocked
```

（任意）NBMAネットワークのLSPフラッディングを、高度にメッシュ化されたポイントツーポイント トポロジで最適化します。

- このコマンドは高度にメッシュ化されたポイントツーポイント トポロジの NBMA ネットワークのみに適しています。

ステップ 15 **commit**

ステップ 16 **show isis interface** [*type interface-path-id* | **level** { **1** | **2** }] [**brief**]

例：

```
RP/0/RP0/CPU0:router# show isis interface HundredGigE 0/9/0/0 brief
```

（任意）IS-IS インターフェイスに関する情報を表示します。

ステップ 17 **show isis** [**instance** *instance-id*] **database** [**level** { **1** | **2** }] [**detail** | **summary** | **verbose**] [* | *lsp-id*]

例：

```
RP/0/RP0/CPU0:router# show isis database level 1
```

（任意）IS-IS LSP データベースを表示します。

ステップ 18 **show isis** [**instance** *instance-id*] **lsp-log** [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router# show isis lsp-log
```

（任意）LSP ログ情報を表示します。

ステップ 19 **show isis database-log** [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router# show isis database-log level 1
```

（任意）IS-IS データベース ログ情報を表示します。

残りの最小ライフタイム

残りの最小ライフタイム機能は、LSPの早期消去と不要なフラッディングを防止します。残りのライフタイムフィールドがフラッディング中に破損した場合、この破損は検出されません。このような破損の結果は、残りのライフタイム値がどのように変更されるかによって異なります。この機能により、IS-IS で受信した LSP の残りのライフタイム値を LSP 最大ライフタイムにリセットすることで、この問題が解決されます。デフォルトでは、LSP最大ライフタイムは 1200 秒に設定され、**max-lsp-lifetime** *seconds* コマンドを使用すると異なる値を設定できます。このアクションによって、受信した残りのライフタイムがどんな値であっても、LSP最大ライフタイムまで LSP がデータベース内に存在する限り、LSP の発信元以外のシステムが LSP を消去しないことが保障されます。

LSP の残りのライフタイムが 0 に達すると、LSP はリンク ステート データベースでさらに 60 秒間保持されます。この追加のライフタイムはゼロ エージング ライフタイムと呼ばれます。ゼロ エージング ライフタイムが経過しても、対応するルータが LSP を更新しない場合、LSP はリンク ステート データベースから削除されます。

また、残りのライフタイムフィールドはネットワークの問題を特定する場合にも役立ちます。受信した LSP のライフタイム値がゼロ エージング ライフタイム（60 秒）未満の場合、IS-IS は破損したライフタイム イベントであることを示すエラー メッセージを生成します。エラー メッセージの例は次のとおりです。

```
Dec 14 15:36:45.663 : isis[1011]: RECV L2 LSP 1111.1111.1112.03-00 from 1111.1111.1112.03:

 possible corrupted lifetime 59 secs for L2 lsp 1111.1111.1112.03-00 from SNPA
02e9.4522.5326 detected.
```

IS-IS は LSP データベースに受信した残りのライフタイム値を保存します。値は、**Rcvd** フィールド内の **show isis database** コマンド出力に表示されます。

IS-IS 認証

隣接関係の確立を制限するために、hello-password コマンドを使用して認証ができます。また、LSP の交換を制限するために、lsp-password コマンドを使用して認証ができます。

IS-IS はプレーン テキスト認証をサポートしますが、この認証は、無許可のユーザに対するセキュリティを提供しません。プレーンテキスト認証ではパスワードが設定でき、無許可のネットワーキング デバイスがルータと隣接関係を形成することを防ぐことができます。このパスワードはプレーンテキストで交換されるため、IS-IS パケットを表示できるエージェントによって参照される可能性があります。

HMAC-MD5 パスワードが設定されている場合、パスワードはネットワークを介して送信されず、代わりに交換データの完全性を確認するための暗号化チェックサムを計算するために使用されます。

IS-IS では、設定されたパスワードを単純な暗号を使用して保存します。ただし、プレーン テキスト形式のパスワードが、LSP、Sequence Number Protocol（SNP）、hello パケットで使用され、IS-IS パケットを表示するプロセスによって参照される可能性があります。パスワードはプレーン テキスト（クリア テキスト）形式または暗号化形式で入力できます。

ドメイン パスワードを設定するには、レベル 2 で lsp-password コマンドを設定します。エリア パスワードを設定するには、レベル 1 で lsp-password コマンドを設定します。

キーチェーン機能によって、IS-IS で設定済みのキーチェーンを参照できます。IS-IS キーチェーンは、hello および LSP のキーチェーン認証をイネーブルにします。キーチェーンは、IS-IS 内のルータ レベル（lsp-password コマンドの場合）およびインターフェイス レベル（hello-password コマンドの場合）で設定できます。これらのコマンドでは、グローバル キーチェーン設定を参照して、設定されているキーチェーンのグローバル セットからセキュリティ パラメータを取得するように IS-IS プロトコルに指示します。

IS-IS はキーチェーンを使用して、認証のためにヒットレス キー ロールオーバーを実装できます。キー ロールオーバーの仕様は時間に基づき、ピア間に時計の誤差が発生すると、ロールオーバー プロセスが影響を受けます。許容値の指定を設定できるため、承認時間枠をその分だけ（前後に）拡張できます。この承認時間枠により、アプリケーション（ルーティング プロトコルおよび管理プロトコルなど）のヒットレス キー ロールオーバーが容易になります。

IS-IS の認証の設定

ここでは、IS-IS の認証の設定方法について説明します。このタスクはオプションです。

手順

ステップ 1 **configure**

ステップ 2 **router isis** *instance-id*

例：

```
RP/0/RP0/CPU0:router(config)# router isis isp
```

指定したルーティング インスタンスの IS-IS ルーティングをイネーブルにし、ルータをルータ コンフィギュレーション モードにします。

- **is-type** コマンドを使用して、特定のルーティング インスタンスによって実行されるルーティングのレベルを変更できます。

ステップ 3 **lsp-password** { **hmac-md5** | **text** } { **clear** | **encrypted** } *password* [**level** { **1** | **2** }] [**send-only**] [**snp send-only**]

例：

```
RP/0/RP0/CPU0:router(config-isis)# lsp-password hmac-md5 clear password1 level 1
```

LSP 認証パスワードを設定します。

- **hmac-md5** キーワードは、パスワードが HMAC-MD5 認証で使用されることを指定します。
- **text** キーワードは、パスワードがクリアテキスト パスワード認証で使用されることを指定します。
- **clear** キーワードは、入力時にパスワードが暗号化されないことを指定します。
- **encrypted** キーワードは、パスワードが入力時に双方向アルゴリズムを使用して暗号化されていることを指定します。
- **level 1** キーワードは、エリア内の認証のパスワードを設定します（レベル 1 LSP と SNP レベル）。
- **level 2** キーワードは、バックボーン（レベル 2 エリア）の認証パスワードを設定します。
- **send-only** キーワードは、LSP とシーケンス番号プロトコル データ ユニット（SNP）の送信時にこれらに認証を追加します。受信 LSP または SNP は認証されません。
- **snp send-only** キーワードは SNP の送信時に SNP に認証を追加します。受信 SNP は認証されません。

（注） SNP パスワード チェックをディセーブルにするには、 **snp send-only** キーワードを **lsp-password** コマンドで指定する必要があります。

ステップ 4 **interface** *type interface-path-id*

例：

```
RP/0/RP0/CPU0:router(config-isis)# interface GigabitEthernet 0/0/0/3
```

インターフェイス設定モードを開始します。

ステップ 5 **hello-password** { **hmac-md5** | **text** } { **clear** | **encrypted** } *password* [**level** { **1** | **2** }] [**send-only**]

例：

```
RP/0/RP0/CPU0:router(config-isis-if)#hello-password text clear mypassword
```

IS-IS インターフェイスの認証パスワードを設定します。

ステップ 6 **commit**

IS-IS のキーチェーンの設定

ここでは、IS-IS のキーチェーンの設定方法について説明します。このタスクはオプションです。

キーチェーンは IS-IS 内のルータ レベル（**lsp-password** コマンド）およびインターフェイス レベル（**hello-password** コマンド）で設定できます。これらのコマンドでは、グローバル キーチェーン設定を参照して、設定されているキーチェーンのグローバルセットからセキュリティ パラメータを取得するように IS-IS プロトコルに指示します。ルータ レベルの設定（lsp-password コマンド）では、ルータで生成されるすべての IS-IS LSP と、すべての Sequence Number Protocol Data Unit（SN PDU）でキーチェーンを使用するように設定します。HELLO PDU で使用されるキーチェーンはインターフェイス レベルで設定され、IS-IS が設定されたインターフェイスごとに異なる値を設定できます。

手順

ステップ 1 **configure**

ステップ 2 **router isis** *instance-id*

例：

```
RP/0/RP0/CPU0:router(config)# router isis isp
```

指定したルーティング インスタンスの IS-IS ルーティングをイネーブルにし、ルータをルータ コンフィギュレーション モードにします。

- **is-type** コマンドを使用して、特定のルーティング インスタンスによって実行されるルーティングのレベルを変更できます。

ステップ 3 **l sp-password keychain** *keychain-name* [**level** { **1** | **2** }] [**send-only**] [**snp send-only**]

例：

```
RP/0/RP0/CPU0:router(config-isis)# lsp-password keychain isis_a level 1
```

キーチェーンを設定します。

ステップ 4 **interface** *type interface-path-id*

例：

```
RP/0/RP0/CPU0:router(config-isis)# interface HundredGigE 0/9/0/0
```

インターフェイス設定モードを開始します。

ステップ 5 **hello-password keychain** *keychain-name* [**level** { **1** | **2** }] [**send-only**]

例：

```
RP/0/RP0/CPU0:router(config-isis-if)#hello-password keychain isis_b
```

IS-IS インターフェイスの認証パスワードを設定します。

ステップ 6 **commit**

ノンストップ フォワーディング

Cisco IOS XR ソフトウェア では、IS-IS NSF により IS-IS プロセスの再起動後にユーザがネットワークにアクセスできない時間が最小限になります。

IS-IS プロセスが再起動すると、そのデバイスのすべてのルーティング ピアは、デバイスがダウンし、その後再びアップになったことを検知します。このような移行によって、いわゆるルーティング フラップが発生します。ルーティング フラップは、複数のルーティング ドメインに広がる場合があります。ルーティングの再起動によって発生したルーティング フラップによって、ルーティングが不安定になります。これはネットワーク全体のパフォーマンスに悪影響を及ぼします。NSF はルーティング フラップを抑止することによって、ネットワークの安定性を保ちます。

NSF では、プロセスの再起動後にルーティング プロトコル情報を復元する一方で、データ パケットの転送を既知のルートで継続できます。NSF 機能が設定されると、ピア ネットワーキング デバイスではルーティング フラップが発生しません。RP フェールオーバー イベント間のルーティングを維持するには、NSF に加えて NSR を設定する必要があります。

IS-IS ルーティングを実行している Cisco IOS XR ルータがプロセスの再起動を行うときは、リンクステート データベースを IS-IS ネイバーと再同期するために、2 つのタスクをルータが実行する必要があります。まず、ネイバー関係をリセットせずに、ネットワーク上の使用可能な IS-IS ネイバーを再学習します。次に、ルータはネットワークのリンクステート データベースのコンテンツを再取得します。

NSF を設定する場合、IS-IS NSF 機能には次の 2 つのオプションがあります。

- IETF NSF
- Cisco NSF

ネットワーク セグメント上の隣接ルータが NSF 対応の場合、つまり隣接ルータが RFC5306 をサポートするソフトウェア バージョンを実行している場合、それらのルータは、**nsf ietf** コマンドで設定されたルータの再起動をサポートします。IETF NSF を使用すると、隣接ルータは、

フェールオーバー後のルーティング情報を再構築するための隣接情報およびリンク ステート情報を提供します。

Cisco IOS XR ソフトウェア では、Cisco NSF が再起動からの回復に必要なすべての状態をチェックポイントで（永続的に）保存し、隣接ルータからの特別な協力を必要としません。状態は隣接ルータによって回復されますが、IS-IS ルーティング プロトコルの標準機能のみを使用します。この機能により Cisco NSF は、他のルータが IETF 標準の NSF 実装を使用していないネットワークでの使用に適しています。

（注） IETF NSF を Cisco IOS XR ルータで設定し、隣接ルータが IETF NSF をサポートしていない場合には、隣接はフラップの影響を受けますが、IETF NSF をサポートしているすべてのネイバーではノンストップ フォワーディングが維持されます。IETF NSF をサポートするネイバーがない場合は、再起動はコールド スタートになります。

IS-IS のノンストップ フォワーディングの設定

ここでは、ルータに NSF を設定する方法について説明します。NSF は、ソフトウェアがプロセスの再起動後に IS-IS リンクステート データベースを IS-IS ネイバーと再同期できるようにします。プロセスは次の原因で再起動する可能性があります。

- RP フェールオーバー（ウォーム リスタートのため）
- 単純なプロセスの再起動（IS-IS のリロードなどの管理要求によるプロセスの再起動）
- IS-IS のソフトウェア アップグレード

いずれの場合でも、NSF はリンク フラップおよびユーザ セッションの損失を低減します。このタスクはオプションです。

手順

ステップ 1 **configure**

ステップ 2 **router isis** *instance-id*

例：

```
RP/0/RP0/CPU0:router(config)# router isis isp
```

指定したルーティング インスタンスの IS-IS ルーティングをイネーブルにし、ルータをルータ コンフィギュレーション モードにします。

- **is-type** ルータ コンフィギュレーション コマンドを使用して、特定のルーティング インスタンスによって実行されるルーティングのレベルを変更できます。

ステップ 3 **nsf** { **cisco** | **ietf** }

例：

```
RP/0/RP0/CPU0:router(config-isis)# nsf ietf
```

次の再起動で NSF をイネーブルにします。

- NSF 対応ネットワーキング デバイスが隣接していない可能性がある異種ネットワークで IS-IS を実行するには、 **cisco** キーワードを入力します。
- 隣接するすべてのネットワーキング デバイスが IETF ドラフトベースの再起動性をサポートする同種ネットワークで IS-IS をイネーブルにするには、**ietf** キーワードを入力します。

ステップ 4 **nsf interface-expires** *number*

例：

```
RP/0/RP0/CPU0:router(config-isis)# nsf interface-expires 1
```

確認された NSF の再開確認応答を再送信する回数を設定します。

- NSF 再起動の間に再送上限数に達した場合、再起動はコールド リスタートになります。

ステップ 5 **nsf interface-timer** *seconds*

例：

```
RP/0/RP0/CPU0:router(config-isis) nsf interface-timer 15
```

各再起動応答を待機する秒数を設定します。

ステップ 6 **nsf lifetime** *seconds*

例：

```
RP/0/RP0/CPU0:router(config-isis)# nsf lifetime 20
```

NSF 再開に続くルートの最大有効期間を設定します。

- この設定時間は再起動の最中にルーティング情報ベース（RIB）にルートを維持する時間であるため、このコマンドには NSF 再起動全体の実行に必要な時間を設定します。
- 設定する値が大きすぎると、古いルートが残ります。
- 設定する値が小さすぎると、ルートの破棄が早すぎる結果になります。

ステップ 7 **commit**

ステップ 8 **show running-config** [*command*]

例：

```
RP/0/RP0/CPU0:router# show running-config router isis isp
```

（任意）現在の実行コンフィギュレーション ファイルの内容全体またはファイルのサブセットを表示します。

- NSF 対応デバイスの IS-IS 設定に「nsf」が表示されていることを確認します。
- この例では、コンフィギュレーション ファイルの内容の「isp」インスタンスのみを示しています。

ISIS NSR

ノンストップ ルーティング（NSR）は、プロセッサのスイッチオーバー イベント（RP フェールオーバーまたは ISSU）中に冗長なルート プロセッサを持つデバイスの IS-IS ルーティングの変更を抑制し、ネットワークの不安定性とダウンタイムを低減します。ノンストップ ルーティングが使用されている場合、アクティブからスタンバイ RP への切り替えは、ネットワーク内の他の IS-IS ルータには影響しません。ルーティング プロトコルのピアリング状態を継続するのに必要なすべての情報は、スイッチオーバー前にスタンバイ プロセッサに転送されるため、スイッチオーバー直後に処理を続行できます。

プロセスの再起動間のルーティングを維持するには、NSR に加えて NSF を設定する必要があります。

ISIS-NSR の設定

手順

ステップ 1 **configure**

ステップ 2 **router isis** *instance-id*

例：

```
RP/0/RP0/CPU0:router(config)# router isis 1
```

指定したルーティング インスタンスの IS-IS ルーティングをイネーブルにし、ルータをルータ コンフィギュレーション モードにします。

ステップ 3 **nsr**

例：

```
RP/0/RP0/CPU0:router(config-isis)# nsr
```

NSR 機能を設定します。

ステップ 4 **commit**

ステップ 5 **show isis nsr adjacency**

例：

```
RP/0/RP0/CPU0:router
# show isis nsr adjacency
System Id Interface SNPA State Hold Changed NSF IPv4 BFD IPv6 BFD
  R1-v1S   Nii0     *PtoP* Up   83  00:00:33 Yes  None    None
```

隣接関係情報を表示します。

ステップ 6 **show isis nsr status**

例：

```
RP/0/RP0/CPU0:router
router#show isis nsr status
IS-IS test NSR(v1a) STATUS (HA Ready):
                           V1 Standby V2 Active V2 Standby
SYNC STATUS:                  TRUE      FALSE(0) FALSE(0)
PEER CHG COUNT:                1           0        0
UP TIME:                      00:03:12     not up    not up
```

NSR のステータス情報を表示します。

ステップ 7 **show isis nsr statistics**

例：

```
RP/0/RP0/CPU0:router
router#show isis nsr statistics
IS-IS test NSR(v1a) MANDATORY STATS :
                                  V1 Active                 V1 Standby          V2 Active
          V2 Standby
L1 ADJ:                                0                     0                     0
               0
L2 ADJ:                                2                     2                     0
               0
LIVE INTERFACE:                        4                     4                     0
               0
PTP INTERFACE:                         1                     1                     0
               0
LAN INTERFACE:                         2                     2                     0
               0
LOOPBACK INTERFACE:                    1                     1                     0
               0
TE Tunnel:                             1                     1                     0
               0
TE LINK:                               2                     2                     0
               0
NSR OPTIONAL STATS :
L1 LSP:                                0                     0                     0
               0
L2 LSP:                                4                     4                     0
               0
IPV4 ROUTES:                           3                     3                     0
               0
IPV6 ROUTES:                           4                     4                     0
               0
```

アクティブ ルータおよびスタンバイ ルータ上の ISIS 隣接関係、lsp、ルート、トンネル、Te リンクの数を示します。

マルチプロトコル ラベル スイッチング トラフィック エンジニアリング

MPLS TE 機能を使用すると、MPLS バックボーンで、レイヤ 2 ATM およびフレーム リレー ネットワークが持つトラフィック エンジニアリングの能力を再現し、そのうえで機能を拡張することができます。MPLS は、レイヤ 2 テクノロジーとレイヤ 3 テクノロジーを統合したものです。

IS-IS では、MPLS TE はリソース予約プロトコル（RSVP）を使用して自動的にバックボーン全体に MPLS TE ラベルスイッチド パスを確立して維持します。ラベルスイッチド パスが使用するルートは、ラベルスイッチド パスのリソース要件とネットワーク リソース（帯域幅など）によって決定されます。利用可能なリソースは、IS-IS の特別な IS-IS TLV 拡張を使用してフラッディングされます。ラベルスイッチド パスは明示的なルートであり、トラフィック エンジニアリング（TE）トンネルとして参照されます。

IS-IS の MPLS トラフィック エンジニアリングの設定

このタスクでは、MPLS TE の IS-IS を設定する手順について説明します。このタスクはオプションです。

始める前に

ルータで IS-IS の MPLS TE をイネーブルにする前に、ネットワークで MPLS ソフトウェア機能をサポートする必要があります。

（注） ネットワークのトラフィック エンジニアリング部分にあるすべての IS-IS ルータ上で、次のタスク リストのコマンドを入力する必要があります。

（注） MPLS トラフィック エンジニアリングでは、現在、番号なし IP リンクを介したルーティングおよびシグナリングはサポートされていません。このため、このようなリンク上には、この機能を設定しないでください。

手順

ステップ 1 **configure**

ステップ 2 **router isis** *instance-id*

例：

```
RP/0/RP0/CPU0:router(config)# router isis isp
```

指定したルーティング インスタンスの IS-IS ルーティングをイネーブルにし、ルータをルータ コンフィギュレーション モードにします。

- **is-type** ルータ コンフィギュレーション コマンドを使用して、特定のルーティング インスタンスによって実行されるルーティングのレベルを変更できます。

ステップ 3 **address-family** { **ipv4** | **ipv6** } [**unicast**]

例：

```
RP/0/RP0/CPU0:router(config-isis)#address-family ipv4 unicast
```

IPv4 または IPv6 アドレス ファミリを指定して、ルータ アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 4 **mpls traffic-eng level** { **1** | **2** }

例：

```
RP/0/RP0/CPU0:router(config-isis-af)# mpls traffic-eng level 1
```

指定した IS-IS レベルに MPLS TE リンク情報をフラッディングするように IS-IS を実行するルータを設定します。

ステップ 5 **mpls traffic-eng router-id** { *ip-address* | *interface-name interface-instance* }

例：

```
RP/0/RP0/CPU0:router(config-isis-af)# mpls traffic-eng router-id loopback0
```

ノードの MPLS TE ルータ ID を指定した IP アドレスまたは指定したインターフェイスに関連付けられている IP アドレスにするように指定します。

ステップ 6 **metric-style wide** [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis-af)# metric-style wide level 1
```

レベル 1 エリアでワイド リンク メトリックのみを生成して受け入れるようにルータを設定します。

ステップ 7 **commit**

ステップ 8 **show isis** [**instance** *instance-id*] **mpls traffic-eng tunnel**

例：

```
RP/0/RP0/CPU0:router# show isis instance isp mpls traffic-eng tunnel
```

（任意）MPLS TE トンネル情報を表示します。

ステップ 9 **show isis** [**instance** *instance-id*] **mpls traffic-eng adjacency-log**

例：

```
RP/0/RP0/CPU0:router# show isis instance isp mpls traffic-eng adjacency-log
```

（任意）MPLS TE IS-IS 隣接変更のログを表示します。

ステップ 10 **show isis** [**instance** *instance-id*] **mpls traffic-eng advertisements**

例：

```
RP/0/RP0/CPU0:router# show isis instance isp mpls traffic-eng advertisements
```

（任意）MPLS TE から最後にフラッディングされた記録を表示します。

MPLS TE 転送隣接

MPLS TE 転送隣接により、ネットワーク管理者はトラフィック エンジニアリングおよびラベル スイッチ パス（LSP）トンネルを、Shortest Path First（SPF）アルゴリズムに基づいた内部ゲートウェイ プロトコル（IGP）ネットワーク内のリンクとして処理できます。転送隣接は、同じ IS-IS レベルのルータ間で作成できます。ルータとルータは、間に何個かホップを入れて配置できます。この結果、TE トンネルに関連付けられたリンク コストで、IGP ネットワーク内のリンクとして、アドバタイズされます。TE ドメインの外側にあるルータは、TE トンネルを参照し、その TE トンネルを使用して、ネットワーク内でトラフィックをルーティングするための最短パスを計算します。

MPLS TE 転送隣接は、双方向接続性確認に成功した場合のみ IS-IS SPF で考慮されます。これには転送隣接が双方向であるか、または MPLS TE トンネルのヘッドエンドとテール エンドのルータが隣接している場合が該当します。

MPLS TE 転送隣接機能は、IS-IS でサポートされます。MPLS TE 転送隣接機能の設定の詳細については、『MPLS Configuration Guide』を参照してください。

IS-IS の隣接関係の調整

このタスクでは、隣接状態変更のロギングをイネーブルにし、IS-IS 隣接パケットのタイマーを変更して、隣接状態のさまざまな側面を表示する方法について説明します。IS-IS 隣接を調整することにより、リンクで輻輳が発生している場合のネットワークの安定性が向上します。このタスクはオプションです。

ポイントツーポイント リンクの場合、IS-IS はレベル 1 とレベル 2 に対して単一の hello だけを送信します。これは、ポイントツーポイント リンクでの level 修飾子が無意味であることを意味します。ポイントツーポイント インターフェイスの hello パラメータを変更するには、level オプションの指定を省略します。

インターフェイス サブモードで設定可能なオプションは、そのインターフェイスだけに適用されます。デフォルトで、値はレベル 1 とレベル 2 に適用されます。

hello-password コマンドを使用して無許可のルータや望ましくないルータとの隣接関係の形成を防ぐことができます。この機能は、隣接関係の確立が望ましくないルータとの接続が多く見られる LAN では特に有効です。

手順

ステップ 1 **configure**

ステップ 2 **router isis** *instance-id*

例：

```
RP/0/RP0/CPU0:router(config)# router isis isp
```

指定したルーティング インスタンスの IS-IS ルーティングをイネーブルにし、ルータをルータ コンフィギュレーション モードにします。

- **is-type** コマンドを使用して、特定のルーティング インスタンスによって実行されるルーティングのレベルを変更できます。

ステップ 3 **log adjacency changes**

例：

```
RP/0/RP0/CPU0:router(config-isis)# log adjacency changes
```

IS-IS の隣接状態の変更時にログ メッセージを生成します（Up または Down）。

ステップ 4 **interface** *type interface-path-id*

例：

```
RP/0/RP0/CPU0:router(config-isis)# interface HundredGigE 0/9/0/0
```

インターフェイス設定モードを開始します。

ステップ 5 **hello-padding** { **disable** | **sometimes** } [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis-if)# hello-padding sometimes
```

ルータの IS-IS インターフェイスの IS-IS hello PDU のパディングを設定します。

- helloパディングはこのインターフェイスのみに適用され、すべてのインターフェイスには適用されません。

ステップ 6 **hello-interval** *seconds* [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis-if)#hello-interval 6
```

ソフトウェアが送信する hello パケット間の時間間隔を指定します。

ステップ 7 **hello-multiplier** *multiplier* [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis-if)# hello-multiplier 10
```

ネイバーが見落とすことができる IS-IS hello パケット数の最大値を指定します。見落とされたパケット数がこの値を超えると、ルータは隣接がダウンしていると宣言します。

- 大きい値にするとネットワークが許容するドロップ パケットの数が増加しますが、隣接ルータの障害の検出に必要な時間も増加します。
- 隣接ルータの障害が検出されないと、逆により多くのパケットが失われる結果になる可能性があります。

ステップ 8 **h ello-password** { **hmac-md5** | **text** } { **clear** | **encrypted** } *password* [**level** { **1** | **2** }] [**send-only**]

例：

```
RP/0/RP0/CPU0:router(config-isis-if)# hello-password text clear mypassword
```

このシステムが hello パケットの認証を含むことを指定し、ネイバーからの hello パケットの認証が成功し、隣接関係が確立することが必要です。

ステップ 9 **commit**

ステップ 10 **show isis** [**instance** *instance-id*] **adjacency** *t ype interface- path-id*] [**detail**] [**systemid** *system-id*]

例：

```
RP/0/RP0/CPU0:router# show isis instance isp adjacency
```

（任意）IS-IS 隣接を表示します。

ステップ 11 show isis adjacency-log

例：

```
RP/0/RP0/CPU0:router# show isis adjacency-log
```

（任意）最新の隣接状態の遷移ログを表示します。

ステップ 12 **show isis** [**instance** *instance-id*] **interface** [*type interface-path-id*] [**brief** | **detail**] [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router# show isis interface HundredGigE 0/9/0/0 brief
```

（任意）IS-IS インターフェイスに関する情報を表示します。

ステップ 13 **show isis** [**instance** *instance-id*] **neighbors** [*interface-type interface-instance*] [**summary**] [**detail**] [**systemid** *system-id*]

例：

```
RP/0/RP0/CPU0:router# show isis neighbors summary
```

（任意）IS-IS ネイバーに関する情報を表示します。

MPLS ラベル配布プロトコル IGP 同期

マルチ プロトコル ラベル スイッチング（MPLS）ラベル配布プロトコル（LDP）内部ゲートウェイ プロトコル（IGP）同期では、IGP パスをスイッチングが使用される前に LDP のラベル交換を完了させることができます。次の 2 つの状況では MPLS のトラフィック損失が発生する可能性があります。

- IGP 隣接が確立されると、LDP がそのリンク ピアとラベルを交換する前に、ルータが新しい隣接を使用してパケットの転送を開始します。
- LDP セッションが閉じられるときに、確立した LDP セッションの代替パスを使用せずに LDP ピアと関連付けられたリンクを使用してルータがトラフィックの転送を続ける。

この機能は、LDP と IS-IS を同期させるメカニズムを提供し、MPLS のパケット損失を最小化します。この同期は、LDP セッションの状態に基づいてネイバーの IS-IS リンクステート パケット（LSP）のリンク メトリックを変更することで実現されます。

リンク上で IS-IS の隣接関係は確立されているが、LDP セッションが失われているかまたは LDP がラベルの交換をまだ完了していないときには、IS-IS は最大のメトリックをそのリンクでアドバタイズします。このインスタンスでは、LDP IS-IS 同期はまだ実現されていません。

（注） IS-IS では、最大のメトリック（0xFFFFFF）を持つリンクは Shortest Path First（SPF）として考慮されません。このため最大のメトリックである -1（0XFFFFFE）が MPLS LDP IGP 同期で使用されます。

LDP IS-IS 同期が達成されると、IS-IS は通常の（設定されたまたはデフォルトの）メトリックをそのリンクでアドバタイズします。

MPLS LDP IS-IS 同期の設定

このタスクは、マルチプロトコルラベルスイッチング（MPLS）ラベル配布プロトコル（LDP）IS-IS 同期をイネーブルにする方法について説明します。MPLS LDP 同期は、インターフェイス コンフィギュレーション モードで、アドレス ファミリに対してイネーブルにすることができます。IPv4 ユニキャスト アドレス ファミリのみがサポートされます。このタスクはオプションです。

手順

ステップ 1 **configure**

ステップ 2 **router isis** *instance-id*

例：

```
RP/0/RP0/CPU0:router(config)# router isis isp
```

指定したルーティング プロセスの IS-IS ルーティングをイネーブルにし、ルータをルータ コンフィギュレーション モードにします。

- 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 とレベル 2 になります。**is-type** コマンドを使用して、特定のルーティング インスタンスによって実行されるルーティングのレベルを変更できます。

ステップ 3 **interface** *type interface-path-id*

例：

```
RP/0/RP0/CPU0:router(config-isis)# interface HundredGigE 0/9/0/0
```

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 4 **address-family ipv4 unicast**

例：

```
RP/0/RP0/CPU0:router(config-isis-if)# address-family ipv4 unicast
```

IPv4 アドレス ファミリを指定し、ルータ アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 5 **mpls ldp sync** [**level** { **1** | **2** }]

例：

```
RP/0/RP0/CPU0:router(config-isis-if-af)# mpls ldp sync level 1
```

インターフェイス HundredGigE 0/9/0/0 の IPv4 アドレス ファミリに対して MPLS LDP 同期をイネーブルにします。

ステップ 6 **commit**

IS-IS 過負荷ビット無効化

IS-IS過負荷ビット無効化機能により、ネットワーク管理者は、ラベル スイッチド パス（LSP）内のルータに Intermediate System-to-Intermediate System（IS-IS）の過負荷ビットが設定されているときにパスがディセーブルになることを防止できます。

IS-IS 過負荷ビット無効化機能がアクティブ化されると、過負荷ビットが設定されているすべてのノード（先頭ノード、中間ノード、終端ノードを含む）は無視されます。つまり、それらはラベル スイッチド パス（LSP）で使用できます。

（注） IS-IS 過負荷ビット無効化機能は、ノードがパス計算（PCALC）に含まれていない場合には、過負荷ビットが設定されたノードのデフォルトの動作を変更しません。

IS-IS 過負荷ビット無効化機能は、次のコマンドでアクティブ化されます。

mpls traffic-eng path-selection ignore overload

IS-IS 過負荷ビット無効化機能は、このコマンドの **no** 形式で非アクティブ化されます。

no mpls traffic-eng path-selection ignore overload

IS-IS 過負荷ビット無効化機能が非アクティブ化されると、過負荷ビットが設定されたノードは最終手段のノードとして使用されません。

IS-IS 過負荷ビット無効化の設定

ここでは、IS-IS 過負荷ビット無効化をアクティブにする方法について説明します。

始める前に

IS-IS 過負荷ビット無効化機能は、次の機能をサポートするネットワークでのみ有効です。

- MPLS
- IS-IS

手順

ステップ 1 **configure**

ステップ 2 **mpls traffic-eng path-selection ignore overload**

例：

```
RP/0/RP0/CPU0:router(config)# mpls traffic-eng path-selection ignore overload
```

IS-IS 過負荷ビット無効化をアクティブにします。

IS-IS 過負荷ビット無効化の設定：例

次に、IS-IS 過負荷ビット無効化をアクティブにする例を示します。

```
config
 mpls traffic-eng path-selection ignore overload
```

次に、IS-IS 過負荷ビット無効化を非アクティブにする例を示します。

```
config
 no mpls traffic-eng path-selection ignore overload
```

IS-IS の参照

この項では、IS-IS 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。

IS-IS 機能の概要

小規模の IS-IS ネットワークは、一般的にネットワーク内にすべてのルータが含まれる単一のエリアとして構築されます。ネットワークの規模が大きくなるにしたがって、このネットワークは、すべてのエリアに属する、接続されたすべてのレベル 2 ルータのセットから構成されるバックボーン エリア内に再編成され、その後、このネットワークはローカル エリアに接続されます。ローカル エリア内部では、すべてのルータがすべてのシステム ID に到達する方法を認識しています。エリア間では、ルータはバックボーンへの到達方法を認識しており、バックボーン ルータは他のエリアに到達する方法を認識しています。

IS-IS ルーティング プロトコルは、バックボーンのレベル 2 とレベル 1 エリアの構成、および必要とされるエリア間のルーティング情報の移動をサポートします。ルータはレベル 1 隣接を確立して、ローカルエリア内でルーティングを実行します（エリア内ルーティング）。ルータはレベル 2 隣接を確立して、レベル 1 エリア間でルーティングを実行します（エリア間ルーティング）。

各 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 つずつサポートできます。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。特定のルーティング インスタンスによって実行されるルーティングのレベルを変更するには、is-type コマンドを使用します。

機能制限

IS-IS の複数のインスタンスが実行されている場合、インターフェイスは 1 インスタンス（プロセス）だけに関連付けることができます。インスタンスは、インターフェイスを共有できません。

デフォルト ルート

デフォルト ルートを IS-IS ルーティング ドメインに強制することができます。IS-IS ルーティング ドメインへのルートの再配布を明確に設定しても、デフォルトではソフトウェアが 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。**default-information originate** コマンドを使用すると、IS-IS にデフォルト ルートが生成され、ルート ポリシーで制御できます。ルート ポリシーを使用してデフォルト ルートが通知されるレベルを決定できます。また、ルート ポリシーによって設定できる他のフィルタリング オプションを指定できます。ルート ポリシーを使用することにより、ルータのルーティング テーブル内での他のルートの存在に応じて、デフォルト ルートを条件付きでアドバタイズできます。

ルータの過負荷ビット

過負荷ビットはステート情報の固有ビットであり、ルータの LSP に含まれます。ルータにこのビットが設定されると、このルータがトラフィックの中継に利用できないことがエリア内のルータに通知されます。この機能は次の 4 つの状況で役立ちます。

1. 深刻だが致命的ではないエラーの発生中（メモリ不足など）。
2. プロセスの起動中および再起動中。ルーティング プロトコルが収束するまで過負荷ビットを設定できます。ただし通常の NSF 再起動またはフェールオーバーの最中は使用しません。使用するとルーティング フラップの原因になります。
3. 新しいルータの試験的な導入の最中。導入が検証されるまで過負荷ビットを設定できます。検証後ビットを消去します。
4. ルータのシャットダウン中。ルータのサービスを停止する前に、トポロジからルータを削除するために過負荷ビットを設定できます。

マルチトポロジ動作中の過負荷ビット設定

過負荷ビットは、シングル トポロジの転送に適用されるため、マルチトポロジ操作中に IPv4 および IPv6 に別々に設定およびクリアされる場合があります。このため、過負荷は、ルータ アドレス ファミリ コンフィギュレーション モードで設定されます。IPv4 過負荷ビットが設定されると、エリア内のすべてのルータは、IPv4 の中継トラフィックにこのルータを使用しません。ただし、引き続き IPv6 の中継トラフィックにはこのルータを使用できます。

IS-IS インスタンスの attached ビット

attached ビットは is-type コマンドと **level-1-2** キーワードでルータに設定します。attached ビットはルータが他のエリアに接続されていることを示します（通常はバックボーン経由）。この機能は、ルータがバックボーンへのデフォルトルートとして領域のレベル1ルータから使用できることを意味します。attached ビットは通常、ルータが他のエリアを検出時にレベル2の SPF ルートを計算する間に自動的に設定されます。このビットはルータがバックボーンから切断されると自動的に消去されます。

（注） レベル 2 インスタンスの接続が失われた場合、レベル 1 インスタンスの LSP 内の attached ビットによってレベル2インスタンスへのトラフィックの送信が続けられ、トラフィックのドロップを発生させます。

level-1-2 キーワードの機能を表すために複数のプロセスを使用するときにこの動作をシミュレートするには、レベル 1 プロセスの attached ビットを手動で設定します。

ルート タグの IS-IS サポート

ルート タグの IS-IS サポート機能によって、IS-IS ルート プレフィックスとタグを関連付けてアドバタイズする機能が提供されます。また、この機能により、RIB内のルートプレフィックスのインストール順序のプライオリティ付けを、ルートのタグに基づいて行うことができます。ルート タグはまた、ルート ポリシーでルート プレフィックスの照合に使用される可能性があります（たとえば、再配布に特定のルート プレフィックスを選択する場合）。

特定のインターフェイスでのフラッディングのブロック

この手法では、特定のインターフェイスでLSPフラッディングの使用がブロックされますが、残りのインターフェイスではフラッディングに関して通常どおり動作します。この手法は理解しやすく設定も容易ですが、長期的にはメッシュ グループに比べて維持が難しく、エラーが起こりやすくなります。IS-IS で使用するフラッディング トポロジは、制限するのではなく詳細に調整します。トポロジの制限が多すぎると（多くのインターフェイスをブロックしすぎると）障害時にネットワークの信頼性が失われます。トポロジの制限が少なすぎると（ブロックするインターフェイスが少なすぎると）望ましいスケーラビリティが達成できなくなります。

ブロックされていないすべてのインターフェイスでドロップする場合にネットワークの堅牢性を高めるには、インターフェイス コンフィギュレーション モードで csnp-interval コマンドを使用して、ブロックされているポイントツーポイントリンクで定期的にComplete Sequence Number PDU（CSNP）パケットが使用されるようにします。定期的な CSNP によって、ネットワークの同期が可能になります。

最大 LSP ライフタイムおよび更新間隔

デフォルトでは、ルータは定期的な LSP 更新を 15 分ごとに送信します。LSP はデフォルトで 20 分間、データベースに残ります。そのときまでにリフレッシュされない場合、削除されま

す。LSP 更新間隔、または最大 LSP ライフタイムを変更できます。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。設定された更新間隔がない場合、LSP のタイムアウトを防止するために、必要により LSP 更新間隔がソフトウェアによって調整されます。

メッシュ グループの設定

メッシュ グループ（ルータのインターフェイスのセット）を設定すると、フラッディングを制限できます。特定のメッシュ グループに属するインターフェイスを介して到達可能なすべてのルータには、他のすべてのルータと少なくとも 1 つのリンクがあり、各ルータと緊密に接続されていると見なされます。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 が 1 つのインターフェイスで受信されると、そのルータの他のすべてのインターフェイスでフラッディングされます。メッシュ グループでは、メッシュ グループに属する 1 つのインターフェイスで新しい LSP が受信されると、新しい LSP は、そのメッシュ グループに属する他のインターフェイスではフラッディングされません。

マルチインスタンス IS-IS

最大 5 つの IS-IS インスタンスを構成できます。IS-IS プロセスが異なるインターフェイス セット上で実行されている場合には、複数の IS-IS プロセス上で MPLS を実行できます。各インターフェイスは 1 つの IS-IS インスタンスとだけ関連付けられます。ソフトウェアは、設定時に 2 つのインスタンスによるインターフェイスの二重登録を防止します。2 つの MPLS のインスタンスを設定するとエラーになります。

ルーティング情報ベース（RIB）では、各 IS-IS インスタンスは同じルーティング クライアントとして扱われるため、IS-IS インスタンス間でルートを再配布するときには注意が必要です。RIB ではレベル 1 ルートがレベル 2 ルートよりも優先されることが認識されません。このためレベル 1 とレベル 2 のインスタンスを実行する場合には、2 つのインスタンスに異なるアドミニストレーティブ ディスタンスを設定して強制的に優先する必要があります。

ラベル配布プロトコル IGP 自動設定

ラベル配布プロトコル（LDP）内部ゲートウェイ プロトコル（IGP）自動設定は、IGP インスタンスに使用される一連のインターフェイス上で LDP をイネーブルにする手順を簡素化します。LDP IGP 自動設定は、多数のインターフェイス（LDP がコア内の転送に使用される場合など）および複数の IGP インスタンス上で同時に使用できます。

この機能は、デフォルトの VPN ルーティングおよび転送（VRF）インスタンスとして IPv4 アドレス ファミリをサポートします。

LDP IGP 自動設定は、LDP の個々のインターフェイス上で igp auto-config disable コマンド使用して明示的にディセーブルにすることもできます。これにより LDP は、明示的にディセーブルにされたインターフェイスを除くすべての IGP インターフェイスで受信できます。

LDP IGP 自動設定の設定については、『*MPLS configuration guide*』を参照してください。

LDP グレースフル リスタートとの MPLS LDP-IGP 同期の互換性

LDP グレースフル リスタートは、LDP セッションが失われた場合にトラフィックを保護します。グレースフル リスタートがイネーブルである LDP セッションに障害が発生した場合でも、グレースフル リスタートで保護されている間、インターフェイス上で MPLS LDP IS-IS 同期が実現されます。MPLS LDP IGP 同期は次の状況で最終的に失われます。

- LDP グレースフル リスタートの再接続タイマーが期限切れになる前に、LDP を再起動できない場合。
- LDP グレースフル リスタートの回復タイマーが期限切れになる前に、保護されたインターフェイス上の LDP セッションを回復できない場合。

IGP ノンストップ フォワーディングとの MPLS LDP-IGP 同期の互換性

IS-IS ノンストップ フォワーディング（NSF）は、IS-IS プロセスの再起動中およびルート プロセッサ（RP）のフェールオーバー中にトラフィックを保護します。LDP IS-IS 同期は、インターフェイス上で LDP グレースフル リスタートもイネーブルの場合のみ IS-IS NSF とともにサポートされます。IS-IS NSF がイネーブルでない場合、LDP の同期状態は再起動およびフェールオーバーの際に維持されません。