



IPv6 対応認定

- [IPv6 対応認定の機能履歴 \(1 ページ\)](#)
- [IPv6 対応認定 \(1 ページ\)](#)
- [IPv6 ルート情報の設定 \(2 ページ\)](#)
- [IPv6 ルート情報の確認 \(3 ページ\)](#)

IPv6 対応認定の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

この機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

表 1: IPv6 対応認定の機能履歴

リリース	機能	機能情報
Cisco IOS XE Bengaluru 17.6.1	IPv6 対応認証	この機能は、最新の RFC 仕様に準拠するために必要なさまざまな IPv6 機能を実装することで拡張されています。

IPv6 対応認定

Cisco IOS XE Bengaluru 17.6.1 には、IPv6 対応認定の最新の RFC 仕様に準拠するために必要なさまざまな IPv6 機能が実装されています。新しく実装された IPv6 の機能は次のとおりです。

- **Fragment Processing and Reassembly (RFC8200)** : 最初のフラグメントには、RFC 8200 で指定されているように、最初の上位レベルプロトコル (ULP) ヘッダーまでの必須拡張ヘッダーが含まれている必要があります。
- **Handling Atomic Fragments in Neighbor Discovery (RFC6980)** : 断片化されたネイバー探索パケットは破棄する必要があります。

- **Packet too Big (RFC8201)** : アトミック フラグメンテーションはサポートされていません。IPv6 MTU 要件の 1280 を満たしていないパケットはドロップされます。
- **Route Information Options (RIO) in IPv6 Router Advertisements (RFC4191)** : ルータからホストへの特定のルートを送達するために、新しい RIO が IPv6 ルータ アドバタイズメント メッセージに追加されました。明示的なルート構成により、必要なルートのみがホストにアドバタイズされます。
- **IPv6 Hop-by-Hop Processing (RFC 8200)** : この拡張機能により、ホップバイホップ オプションヘッダー処理を必要とするパケットの配信パスに沿って、ノードを明示的に構成できます。

IPv6 ルート情報の設定

IPv6 ルータ アドバタイズメント メッセージのルート情報オプション (RIO) は、ルータからホストへの特定のルートの通信に役立ちます。そのため、ホストがマルチホーム構成されていて、ルータが異なるリンク上にある場合、ホストが適切なデフォルトルータを選択する機能が向上します。明示的なルート構成により、必要なルートのみがホストにアドバタイズされます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 : Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 2	interface interface 例 : Device(config)# interface gigabitethernet1.1	インターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	ipv6 nd ra specific-route prefix/length lifetime lifetime/infinity [preference preference] 例 : Device(config-if)# ipv6 nd ra specific-route 3::3/116 lifetime 11 preference medium	IPv6 ルータ アドバタイズメント メッセージの RIO を設定します。 詳細については、 ipv6 nd ra specific route コマンドを参照してください。

IPv6 ルート情報の確認

ルーターアダプタイズメントで送信される特定のルートを識別するには、次のコマンドを使用します。

```
Device# show ipv6 nd ra specific-route
```

```
IPv6 Prefix/Length Lifetime Preference Interface
```

```
-----  
1234::12/127 1000 High GigabitEthernet2
```

