



コントロールプレーンポリシングの設定

- [CoPP の制約事項, 1 ページ](#)
- [コントロールプレーンポリシングに関する情報, 2 ページ](#)
- [CoPP の設定方法, 6 ページ](#)
- [CoPP の設定例, 9 ページ](#)
- [CoPP のモニタリング, 11 ページ](#)
- [CoPP に関する追加情報, 12 ページ](#)
- [コントロールプレーンポリシングの機能情報, 13 ページ](#)

CoPP の制約事項

コントロールプレーンポリシング (CoPP) の制約事項は、次のとおりです。

- 入力 CoPP だけがサポートされます。 **system-cpp-policy** ポリシーマップは、入力方向でのみ、コントロールプレーンインターフェイスで使用可能です。
- コントロールプレーンインターフェイスにインストールできるのは、 **system-cpp-policy** ポリシーマップのみです。
- **system-cpp-policy** ポリシーマップおよび 17 個のシステム定義のクラスは、変更または削除することはできません。
- **system-cpp-policy** ポリシーマップの下で許可されるのは、 **police** アクションのみです。さらに、 **police rate** は、パケット/秒単位 (pps) でのみ設定できます。
- 1 つ以上の CPU キューがそれぞれのクラスマップの一部となります。複数の CPU キューが 1 つのクラスマップに属している場合、クラスマップのポリサーレートを変更すると、そのクラスマップに属しているすべての CPU キューに影響します。同様に、クラスマップを無効にすると、そのクラスマップに属するすべてのキューが無効になります。各クラスマップに属する CPU キューの詳細については、 [表 1 : CoPP のシステム定義された値, \(3 ページ\)](#) を参照してください。

コントロールプレーン ポリシングに関する情報

この章では、コントロールプレーン ポリシング (CoPP) がデバイスで機能する仕組みと、それを設定する方法について説明します。

CoPP の概要

CoPP 機能によって、不要なトラフィックまたは DoS トラフィックから CPU を保護し、コントロールプレーンおよび管理トラフィックを優先させることにより、デバイスのセキュリティが向上します。

デバイスは通常、3つの操作プレーンにセグメント化され、それぞれに独自の目的があります。

- データ パケットを転送するための、データ プレーン。
- データを適切にルーティングするための、コントロールプレーン。
- ネットワーク要素を管理するための、管理プレーン。

CoPP を使用することで、大半の CPU 行きトラフィックを保護し、ルーティングの安定性と信頼性を確保し、パケットを確実に配信することができます。特に重要なのは、DoS 攻撃から CPU を保護するために CoPP を使用できることです。

CoPP は、モジュラ QoS コマンドライン インターフェイス (MQC) および CPU キューを使用して、これらの目的を達成します。さまざまなタイプのコントロールプレーントラフィックが特定の条件に基づいてグループ化され、CPU キューに割り当てられます。ハードウェアに専用のポリサーを設定することで、これらの CPU キューを管理できます。たとえば、特定の CPU キュー (トラフィック タイプ) のポリサー レートを変更したり、特定のタイプのトラフィックに対するポリサーを無効にしたりできます。

ポリサーはハードウェアに設定されていますが、CoPP は CPU のパフォーマンスやデータ プレーン のパフォーマンスには影響しません。しかし、CPU に着信するパケット数は制限されるため、CPU 負荷が制御されます。これは、ハードウェアからのパケットを待っているサービスが、より制御された着信パケットのレート (ユーザ設定可能なレート) を確認する可能性があることを意味します。

システム定義の CoPP の特徴

デバイスの初回の電源投入時は、システムによって次のタスクが自動的に実行されます。

- ポリシー マップ **system-cpp-policy** を検索します。このポリシー マップが検出されなかった場合は、ポリシー マップが作成され、コントロールプレーンにインストールされます。
- **system-cpp-policy** の下に 17 のクラス マップを作成します。

次に デバイスの電源を入れたときに、すでに作成済みのポリシーとクラス マップがシステムによって検出されます。

- ポリシーがインストールされると、(32のうち) 16の CPU キューがデフォルトで有効になり、それぞれデフォルトのレートが設定されます。デフォルトでイネーブルになっている CPU キューとそれらのデフォルトのレートを「CoPPのシステム定義値」の表に示します。
- Cisco IOS XE Everest 16.6.1 では、CoPP はデフォルトによりすべてのキューでイネーブルになっています。CPU パントレートとポリサーレートは、system-cpp-policy の下で引き上げる必要があります。

次の表に、デバイスをロードしたときにシステムが作成するクラスマップを示します。各クラスマップに対応するポリサーと、各クラスマップの下にグループ化された1つ以上の CPU キューを示します。クラスマップとポリサーには1対1のマッピングがあり、1つ以上の CPU キューがクラスマップにマッピングします。

表 1: CoPPのシステム定義された値

クラス マップ名	ポリサー インデックス (ポリサー No.)	CPU キュー (キュー No.)	デフォルトでイネーブルになる CPU キュー	デフォルトのポリサーレート: 1秒あたりのパケット数 (pps)
system-cpp-police-data	WK_CPP_POLICE_DATA(0)	WK_CPU_Q_ICMP_GEN(3) WK_CPU_Q_BROADCAST(12)	○	200
system-cpp-police-l2-control	WK_CPP_POLICE_L2_CONTROL(1)	WK_CPU_Q_L2_CONTROL(1)	なし	500
system-cpp-police-routing-control	WK_CPP_POLICE_ROUTING_CONTROL(2)	WK_CPU_Q_ROUTING_CONTROL(4)	○	500
system-cpp-police-control-low-priority	WK_CPP_POLICE_CONTROL_LOW_PRI(3)	WK_CPU_Q_ICMP_REDIRECT(6) WK_CPU_Q_GENERAL_PUNT(25)	なし	500
system-cpp-police-punt-webauth	WK_CPP_POLICE_PUNT_WEBAUTH(7)	WK_CPU_Q_PUNT_WEBAUTH(22)	なし	[1000]
system-cpp-police-topology-control	WK_CPP_POLICE_TOPOLOGY_CONTROL(8)	WK_CPU_Q_TOPOLOGY_CONTROL(15)	なし	13000

クラス マップ名	ポリサー インデックス (ポリサー No.)	CPU キュー (キュー No.)	デフォルトでイネーブルになる CPU キュー	デフォルトのポリサーレート: 1秒あたりのパケット数 (pps)
system-cpp-police-multicast	WK_CPP_POLICE_MULTICAST(9)	WK_CPU_Q_TRANSIT_TRAFFIC(18) WK_CPU_Q_MCAST_DATA(30)	○	500
system-cpp-police-sys-data	WK_CPP_POLICE_SYS_DATA(10)	WK_CPU_Q_LEARNING_CACHE_OVFL(13) WK_CPU_Q_CRYPTIO_CONTROL(23) WK_CPU_Q_EXCEPTION(24) WK_CPU_Q_EGR_EXCEPTION(28) WK_CPU_Q_NFL_SAMPLED_DATA(26) WK_CPU_Q_GOLD_PKT(31) WK_CPU_Q_RPF_FAILED(19)	○	100
system-cpp-police-dot1x-auth	WK_CPP_POLICE_DOT1X(11)	WK_CPU_Q_DOT1X_AUTH(0)	なし	[1000]
system-cpp-police-protocol-snooping	WK_CPP_POLICE_PR	WK_CPU_Q_PROTO_SNOOPING(16)	なし	500
system-cpp-police-sw-forward	WK_CPP_POLICE_SW_FWD(13)	WK_CPU_Q_SW_FORWARDING_Q(14) WK_CPU_Q_SGT_CACHE_FULL(27) WK_CPU_Q_LOGGING(21)	○	[1000]
system-cpp-police-forus	WK_CPP_POLICE_FORUS(14)	WK_CPU_Q_FORUS_ADDR_RESOLUTION(5) WK_CPU_Q_FORUS_TRAFFIC(2)	なし	[1000]
system-cpp-police-multicast-end-station	WK_CPP_POLICE_MULTICAST_SNOOPING(15)	WK_CPU_Q_MCAST_END_STATION_SERVICE(20)	○	2000
system-cpp-default	WK_CPP_POLICE_DEFAULT_POLICER	WK_CPU_Q_DHCP_SNOOPING WK_CPU_Q_SHOW_FORWARD	なし	[1000]

- IOS XE 3.x から IOS XE 16.x.x にアップグレードする場合：
 - アップグレード前のデバイスに **system-cpp-policy** ポリシーがなかった場合、アップグレード時にデフォルトのポリシーが作成されます。
 - アップグレード前のデバイスに **system-cpp-policy** という名前のポリシーがあった場合、アップグレード時にポリシーは再生成されません。 **cpp sytem-default** コマンドを実行し、デフォルトのポリシーを動作させます。
 - 主要なイメージアップグレード時に **cpp sytem-default** コマンドを実行して最新のデフォルトのポリサー レートを取得することをお勧めします。
- IOS XE 16.x.x から IOS XE 3.x にダウングレードする場合：
 - **system-cpp-policy** ポリシーはデバイスで保持されますが、コントロールプレーンにはインストールされません。このポリシーは削除できます。
- IOS XE 16.x.x から IOS XE 3.x にダウングレードし、IOS XE 16.x.x にアップグレードする場合：
 - 3.x にダウングレードした後にポリシーを削除してから 16.x.x にアップグレードする場合、ポリシーはデフォルトの設定で生成されます。
 - 3.x にダウングレードした後にポリシーを削除せず 16.x.x にアップグレードする場合、ポリシーは再生成されません。 **cpp sytem-default** コマンドを実行し、デフォルトのポリシーを動作させます。

ユーザ設定可能な CoPP の特徴

次のタスクを実行して、コントロールプレーン トラフィックを管理できます。

- CPU キューを有効または無効にします。

CPU キューを有効にするには、**system-cpp-policy** ポリシーマップ内で、対応するクラスマップの下にポリサー アクション（パケット/秒単位）を設定します。

CPU キューを無効にするには、**system-cpp-policy** ポリシーマップ内で、対応するクラスマップの下にポリサー アクションを削除します。
- **system-cpp-policy** ポリシーマップ内で、対応するクラスマップの下にポリサー レート アクション（パケット/秒単位）を設定することで、ポリサー レートを変更します。
- グローバル コンフィギュレーション モードで **cpp system-default** コマンドを入力することによって、CPU キューをデフォルト値に設定します。

CoPP の設定方法

CPU キューの有効化またはポリサー レートの変更

CPU キューを有効にし、CPU キューのポリサー レートを変更する手順は、同じです。次の手順に従ってください。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	configureterminal 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 3	policy-map policy-map-name 例： Device(config)# policy-map system-cpp-policy Device(config-pmap)#	ポリシー マップ 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	class class-name 例： Device(config-pmap)# class system-cpp-police-protocol-snooping Device(config-pmap-c)#	クラス アクション コンフィギュレーション モードを開始します。有効にする CPU キューに対応するクラスの名前を入力します。「CoPP のシステム定義値」の表を参照してください。
ステップ 5	police rate レート pps 例： Device(config-pmap-c)# police rate 100 pps	指定したトラフィック クラスに対し、1 秒間に処理される着信パケット数の上限を指定します。 (注) 指定するレートは、指定したクラス マップに属するすべての CPU キューに適用されます。

	コマンドまたはアクション	目的
ステップ 6	service-policy input <i>policy-name</i> 例： Device (config) # control-plane Device (config-cp) # service-policy input system-cpp-policy Device (config-cp) #	system-cpp-policy を FED にインストールします。このコマンドは、FED ポリシーを表示するために必要です。このコマンドを設定しないと、エラーになります。
ステップ 7	end 例： Device (config-pmap-c) # end	特権 EXEC モードに戻ります。
ステップ 8	show running-config begin system-cpp-policy 例： Device# show running-config begin system-cpp-policy	さまざまなトラフィック タイプに設定されたレートを表示します。

CPU キューの無効化

CPU キューを無効にするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	configureterminal 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 3	policy-map <i>policy-map-name</i> 例： Device (config) # policy-map	ポリシー マップ コンフィギュレーションモードを開始します。

	コマンドまたはアクション	目的
	<code>system-cpp-policy</code> Device (config-pmap) #	
ステップ 4	<code>class class-name</code> 例： Device (config-pmap) # <code>class</code> <code>system-cpp-police-protocol-snooping</code> Device (config-pmap-c) #	クラスアクションコンフィギュレーションモードを開始します。無効にする CPU キューに対応するクラスの名前を入力します。「CoPP のシステム定義値」の表を参照してください。
ステップ 5	<code>no police rate</code> レート <code>pps</code> 例： Device (config-pmap-c) # <code>no police</code> <code>rate 100 pps</code>	指定したトラフィッククラスの着信パケットの処理を無効にします。 (注) これにより、指定したクラスマップに属するすべての CPU キューが無効になります。
ステップ 6	<code>end</code> 例： Device (config-pmap-c) # <code>end</code>	特権 EXEC モードに戻ります。
ステップ 7	<code>show running-config begin</code> <code>system-cpp-policy</code> 例： Device # <code>show running-config begin</code> <code>system-cpp-policy</code>	さまざまなトラフィックタイプに設定されたレートを表示します。

すべての CPU キューに対するデフォルトのポリサー レートの設定

すべての CPU キューのポリサー レートをデフォルトのレートに設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	<code>enable</code> 例： Device > <code>enable</code>	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。

	コマンドまたはアクション	目的
ステップ 2	configureterminal 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 3	cpp system-default 例： Device(config)# cpp system-default Defaulting CPP : Policer rate for all classes will be set to their defaults	すべてのクラスのポリサー レートをデフォルトのレートに設定します。
ステップ 4	end 例： Device(config)# end	特権 EXEC モードに戻ります。
ステップ 5	show platform hardware fed switch switch-number qos que stat internal cpu policer 例： Device# show platform hardware fed switch 1 qos que stat internal cpu policer	さまざまなトラフィック タイプに設定されたレートを表示します。

CoPP の設定例

例：CPU キューの有効化または CPU キューのポリサー レートの変更

次の例に、CPU キューを有効にする方法、または CPU キューのポリサー レートを変更する方法を示します。ここでは、class system-cpp-police-protocol-snooping CPU キューが有効になり、ポリサー レートは 100 pps です。

```
Device> enable
Device# configure terminal
Device(config)# policy-map system-cpp-policy
Device(config-pmap)# class system-cpp-police-protocol-snooping
Device(config-pmap-c)# police rate 100 pps
Device(config-pmap-c)# end
```

例：すべてのCPUキューに対するデフォルトのポリサー レートの設定

```
Device# show running-config | begin system-cpp-policy

policy-map system-cpp-policy
class system-cpp-police-data
  police rate 200 pps
class system-cpp-police-sys-data
  police rate 100 pps
class system-cpp-police-sw-forward
  police rate 1000 pps
class system-cpp-police-multicast
  police rate 500 pps
class system-cpp-police-multicast-end-station
  police rate 2000 pps
class system-cpp-police-punt-webauth
class system-cpp-police-l2-control
class system-cpp-police-routing-control
  police rate 500 pps
class system-cpp-police-control-low-priority
class system-cpp-police-wireless-priority1
class system-cpp-police-wireless-priority2
class system-cpp-police-wireless-priority3-4-5
class system-cpp-police-topology-control
class system-cpp-police-dot1x-auth
class system-cpp-police-protocol-snooping
  police rate 100 pps
class system-cpp-police-forus
class system-cpp-default

<output truncated>
```

例：すべてのCPUキューに対するデフォルトのポリサー レートの設定

次に、すべてのCPUキューのポリサー レートをデフォルトに設定し、その後に設定を確認する例を示します。ユーザ定義のポリシーはシステム定義のポリシーの上に適用されます。つまり、ユーザ定義のクラス マップに一致する制御トラフィックは、ユーザ定義のCPP ポリサー クラスでの集約ポリサーによって異なります。ユーザ定義トラフィック クラスの統計情報はバイト単位で報告されます。

```
Device> enable
Device# configure terminal
Device(config)# cpp system-default
Defaulting CPP : Policer rate for all classes will be set to their defaults
Device(config)# end
Device show platform hardware fed switch 1 qos queue stats internal cpu policer

(default) (set)
QId PlcIdx Queue Name Enabled Rate Rate Drop
-----
0 11 DOT1X Auth No 1000 1000 0
1 1 L2 Control No 500 500 0
2 14 Forus traffic No 1000 1000 0
3 0 ICMP GEN Yes 200 200 0
4 2 Routing Control Yes 1800 1800 0
5 14 Forus Address resolution No 1000 1000 0
6 3 ICMP Redirect No 500 500 0
7 6 WLESS PRI-5 No 1000 1000 0
8 4 WLESS PRI-1 No 1000 1000 0
9 5 WLESS PRI-2 No 1000 1000 0
10 6 WLESS PRI-3 No 1000 1000 0
11 6 WLESS PRI-4 No 1000 1000 0
12 0 BROADCAST Yes 200 200 0
13 10 Learning cache ovfl Yes 100 100 0
14 13 Sw forwarding Yes 1000 1000 0
15 8 Topology Control No 13000 13000 0
16 12 Proto Snooping No 500 500 0
```

```

17 16 DHCP Snooping No 1000 1000 0
18 9 Transit Traffic Yes 500 500 0
19 10 RPF Failed Yes 100 100 0
20 15 MCAST END STATION Yes 2000 2000 0
21 13 LOGGING Yes 1000 1000 0
22 7 Punt Webauth No 1000 1000 0
23 10 Crypto Control Yes 100 100 0
24 10 Exception Yes 100 100 0
25 3 General Punt No 500 500 0
26 10 NFL SAMPLED DATA Yes 100 100 0
27 2 SGT Cache Full Yes 1800 1800 0
28 10 EGR Exception Yes 100 100 0
29 16 Show frwd No 1000 1000 0
30 9 MCAST Data Yes 500 500 0
    
```

CoPPのモニタリング

CPU キューのトラフィックタイプやポリサーレート（ユーザが設定したレートやデフォルトのレート）などのポリサー設定を表示するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ1	<p>enable</p> <p>例：</p> <pre>Device> enable</pre>	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ2	<p>show platform software fed switch active qos policy target status</p> <p>例：</p> <pre>Device# show platform software fed switch active policy target statusTCG status summary: Loc Interface IIF-ID Dir State:(cfg,opr) Policy ----- ?:255 Control Plane 0x00000001000001 OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-0 0x00000001000002 OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-1 0x00000001000003 OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-2 0x00000001000004 OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-3 0x00000001000005 OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-4 0x00000001000006 OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-5 0x00000001000007 OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-6 0x00000001000008 OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-7 0x00000001000009 OUT VALID,SET_INHW system-cpp-policy</pre>	ユーザ定義の CoPP ポリシーを表示します。

	コマンドまたはアクション	目的
	?:255 CoPP-Queue-8 system-cpp-policy	0x0000000100000a OUT VALID,SET_INHW
	?:255 CoPP-Queue-9 system-cpp-policy	0x0000000100000b OUT VALID,SET_INHW
	?:255 CoPP-Queue-10 system-cpp-policy	0x0000000100000c OUT VALID,SET_INHW
	?:255 CoPP-Queue-12 system-cpp-policy	0x00000001000019 OUT VALID,SET_INHW
	?:255 CoPP-Queue-13 system-cpp-policy	0x0000000100001a OUT VALID,SET_INHW
	?:255 CoPP-Queue-14 system-cpp-policy	0x0000000100001b OUT VALID,SET_INHW
	?:255 CoPP-Queue-15 system-cpp-policy	0x0000000100001c OUT VALID,SET_INHW
	?:255 CoPP-Queue-16 system-cpp-policy	0x0000000100001d OUT VALID,SET_INHW
	?:255 CoPP-Queue-17 system-cpp-policy	0x0000000100001e OUT VALID,SET_INHW
	?:255 CoPP-Queue-18 system-cpp-policy	0x0000000100001f OUT VALID,SET_INHW
	?:255 CoPP-Queue-19 system-cpp-policy	0x00000001000020 OUT VALID,SET_INHW
	?:255 CoPP-Queue-20 system-cpp-policy	0x00000001000021 OUT VALID,SET_INHW
	?:255 CoPP-Queue-21 system-cpp-policy	0x00000001000022 OUT VALID,SET_INHW
	?:255 CoPP-Queue-22 system-cpp-policy	0x00000001000023 OUT VALID,SET_INHW
	?:255 CoPP-Queue-23 system-cpp-policy	0x00000001000024 OUT VALID,SET_INHW
	?:255 CoPP-Queue-24 system-cpp-policy	0x00000001000025 OUT VALID,SET_INHW
	?:255 CoPP-Queue-25 system-cpp-policy	0x00000001000026 OUT VALID,SET_INHW
	?:255 CoPP-Queue-26 system-cpp-policy	0x00000001000027 OUT VALID,SET_INHW
	?:255 CoPP-Queue-27 system-cpp-policy	0x00000001000028 OUT VALID,SET_INHW
	?:255 CoPP-Queue-28 system-cpp-policy	0x00000001000029 OUT VALID,SET_INHW
	?:255 CoPP-Queue-29 system-cpp-policy	0x0000000100002a OUT VALID,SET_INHW
	?:255 CoPP-Queue-30 system-cpp-policy	0x0000000100002b OUT VALID,SET_INHW
	?:255 CoPP-Queue-31 system-cpp-policy	0x0000000100002c OUT VALID,SET_INHW

CoPPに関する追加情報

関連資料

関連項目	参照先
MQC QoS コマンド、および CoPP show コマンド	Command Reference (Catalyst 9300 Series Switches) 『

MIB

MIB	MIB リンク
本リリースでサポートするすべての MIB	選択したプラットフォーム、Cisco IOS リリース、およびフィチャセットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

テクニカル サポート

説明	リンク
<p>シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンラインリソースを提供しています。</p> <p>お使いの製品のセキュリティ情報や技術情報を入手するために、Product Alert Tool (Field Notice からアクセス)、Cisco Technical Services Newsletter、Really Simple Syndication (RSS) フィードなどの各種サービスに加入できます。</p> <p>シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。</p>	http://www.cisco.com/support

コントロールプレーンポリシーの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレーンで各機能のサポートが導入されたときのソフトウェア リリースのみを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよび Cisco ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 2: コントロールプレーンポリシーの機能情報

機能名	リリース	機能情報
コントロールプレーンポリシー	Cisco IOS XE Everest 16.6.1	<p>CoPP 機能によって、不要なトラフィックまたは DoS トラフィックから CPU を保護し、コントロールプレーンおよび管理トラフィックを優先させることにより、デバイスのセキュリティが向上します。</p> <p>この機能は、次のプラットフォームに実装されていました。</p> <ul style="list-style-type: none">• Cisco Catalyst 9300 シリーズ スイッチ