



コントロールプレーンポリシングの設定

- [CoPP の制約事項 \(1 ページ\)](#)
- [CoPP の概要 \(2 ページ\)](#)
- [CoPP の設定方法 \(14 ページ\)](#)
- [CoPP の設定例 \(18 ページ\)](#)
- [CoPP のモニタリング \(22 ページ\)](#)
- [CoPP の機能履歴 \(22 ページ\)](#)

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。システム定義クラスのポリシングレートは、秒単位のパケット/秒 (pps)
- ポリサーレートを設定する場合、クロック周波数の制限により、一部のクラスで表示されるデフォルトレートと設定レートの値に違いが生じることに注意してください (すべてのクラスにデフォルトレートを設定した場合でも)。詳細については、この章の「ユーザー設定可能な CoPP の特徴」および「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」を参照してください。
- 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、クラスマップでポリサーを無効にすると、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。

す。各クラスマップに属するCPUキューの詳細については、「CoPPのシステム定義値」の表を参照してください。

- システム定義のクラスマップのポリサーを無効にすることはお勧めできません。つまり、**no police rate rate pps** コマンドを設定しないでください。これを行うと、CPU へのトラフィックが多い場合に、システム全体の正常性に影響します。さらに、システム定義のクラスマップのポリサーレートを無効にした場合でも、システム起動プロセスを保護するために、システムはシステムのブートアップ後にデフォルトのポリサーレートに自動的に戻ります。
- `system-cpp` ポリシーの下で設定されたクラスがデフォルト値のままの場合、それらのクラスに関する情報は `show run` コマンドで表示されません。代わりに `show policy-map system-cpp-policy` または `show policy-map control-plane` コマンドを使用します。
引き続き `show run` コマンドを使用して、カスタムポリシーに関する情報を表示できます。
- 大量の CPU バウンドパケットを使用するプロトコルは、同じクラスの他のプロトコルに影響を与える可能性があります。これらのプロトコルの一部は同じポリサーを共有するためです。たとえば、Address Resolution Protocol (ARP) は、`system-cpp-police-forus` クラスの Telnet、Internet Control Message Protocol (ICMP)、SSH、FTP、SNMP などのホストプロトコルの配列と 4000 個のハードウェアポリサーを共有します。ARP ポイズニングまたは ICMP 攻撃が発生すると、ハードウェアポリサーは、4000 パケット/秒を超える着信トラフィックのスロットリングを開始し、CPU とシステムの全体的な完全性を保護します。その結果、ARP および ICMP ホストプロトコルは、同じクラスを共有する他のホストプロトコルとともにドロップされます。
- Cisco IOS XE Fuji 16.8.1a 以降、ユーザー定義のクラスマップの作成はサポートされていません。

CoPP の概要

この章では、コントロールプレーンポリシング (CoPP) が device で機能する仕組みと、その設定方法について説明します。

CoPP の概要

CoPP 機能は、不要なトラフィックおよび DoS 攻撃から CPU を保護する device のセキュリティを向上させます。また、他の優先順位の低い大量のトラフィックによって発生するトラフィックのドロップから、制御および管理トラフィックを保護することもできます。

device 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 (ユーザー設定可能なレート) 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

device 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。見つからない場合、システムはそれを作成してコントロールプレーンにインストールします。
- **system-cpp-policy** の下に 18 個のクラスマップを作成します。
次に device 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
- デフォルトで、すべての CPU キューをそれぞれのデフォルトレートで有効にします。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。

system-cpp-policy ポリシーマップはシステム デフォルト ポリシー マップであり、通常はデバイスのスタートアップ コンフィギュレーションに明示的に保存する必要はありません。ただし、スタンバイデバイスとのバルク同期に失敗すると、コンフィギュレーションがスタートアップ コンフィギュレーションから消去される可能性があります。この場合、手動で **system-cpp-policy** ポリシーマップをスタートアップ コンフィギュレーションに保存する必要があります。 **show running-config** 特権 EXEC コマンドを使用して、保存されていることを確認します。

```
policy-map system-cpp-policy
```

次の表に、device 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。ポリサーへのクラスマップの 1 対 1 のマッピングと、CPU キューへのクラスマップの 1 対多マッピングがあります。

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.)
system-cpp-police-data	WK_CPP_POLICE_DATA(0)	WK_CPU_Q_ICMP_GEN(3) WK_CPU_Q_BROADCAST(12) WK_CPU_Q_ICMP_REDIRECT (6)
system-cpp-police-l2- control	WK_CPP_POLICE_L2_ CONTROL(1)	WK_CPU_Q_L2_CONTROL(1)
system-cpp-police-routing-control	WK_CPP_POLICE_ROUTING_CONTROL(4)	WK_CPU_Q_ROUTING_ CONTROL(4) WK_CPU_Q_LOW_LATENCY (27)
system-cpp-police-punt-webauth	WK_CPP_POLICE_PU NT_WEBAUTH(7)	WK_CPU_Q_PUNT_WEBAUTH(22)
system-cpp-police-topology-control	WK_CPP_POLICE_TOPOLOGY_CONTROL(8)	WK_CPU_Q_TOPOLOGY_CONTROL(15)
system-cpp-police- multicast	WK_CPP_POLICE_MULTICAST(9)	WK_CPU_Q_TRANSIT_ TRAFFIC(18) WK_CPU_Q_MCAST_DATA(30)
system-cpp-police-sys- data	WK_CPP_POLICE_SYS_ DATA(10)	WK_CPU_Q_OPENFLOW (13) WK_CPU_Q_CRYPTO_ 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)
system-cpp-police-dot1x-auth	WK_CPP_POLICE_DOT1X(11)	WK_CPU_Q_DOT1X_AUTH(0)
system-cpp-police- protocol-snooping	WK_CPP_POLICE_PR(12)	WK_CPU_Q_PROTO_ SNOOPING(16)
system-cpp-police-dhcp-snooping	WK_CPP_DHCP_SNOOPING(6)	WK_CPU_Q_DHCP_SNOOPING(17)
system-cpp-police-sw-forward	WK_CPP_POLICE_SW_FWD(13)	WK_CPU_Q_SW_FORWARDING_Q(14) WK_CPU_Q_LOGGING(21) WK_CPU_Q_L2_LVX_DATA_PACK (11)

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.)
system-cpp-police-forus	WK_CPP_POLICE_FORUS(14)	WK_CPU_Q_FORUS_ADDR_RESOLUTION(5) WK_CPU_Q_FORUS_TRAFFIC(2)
system-cpp-police-multicast-end-station	WK_CPP_POLICE_MULTICAST_END_STATION(6)	WK_CPU_Q_MCAST_END_STATION_SERVICE(20)
system-cpp-default	WK_CPP_POLICE_DEFAULT(12)	WK_CPU_Q_INTER_FED_TRAFFIC(7) WK_CPU_Q_EWLC_CONTROL(9) WK_CPU_Q_EWLC_DATA(10)
system-cpp-police-stackwise-virt-control	WK_CPP_STACKWISE_VIRTUAL_CONTROL(16)	WK_CPU_Q_STACKWISE_VIRTUAL_CONTROL(29)
system-cpp-police-l2lvx-control	WK_CPP_L2_LVX_CONT_PACK(4)	WK_CPU_Q_L2_LVX_CONT_PACK(8)
system-cpp-police-high-rate-app	WK_CPP_HIGH_RATE_APP(18)	WK_CPU_Q_HIGH_RATE_APP(23)
system-cpp-police-system-critical	WK_CPP_SYSTEM_CRITICAL(3)	WK_CPU_Q_SYSTEM_CRITICAL(25)

次の表に、CPU キューと、各 CPU キューに関連付けられた機能を示します。

表 2: CPU キューと関連機能

CPU キュー (キュー No.)	機能
WK_CPU_Q_DOT1X_AUTH(0)	IEEE 802.1x ポートベースの認証

CPU キュー (キュー No.)	機能
WK_CPU_Q_L2_CONTROL(1)	ダイナミック トランッキング プロトコル (DTP) VLAN トランッキング プロトコル (VTP) ポート集約プロトコル (PAgP) Client Information Signalling Protocol (CISP) メッセージセッションリレープロトコル マルチ VLAN 登録プロトコル (MVRP) Metropolitan Mobile Network (MMN) リンクレベル検出プロトコル (LLDP) 単一方向リンク検出 (UDLD) リンク集約制御プロトコル (LACP) Cisco Discovery Protocol (CDP) スパニング ツリー プロトコル (STP)
WK_CPU_Q_FORUS_TRAFFIC(2)	Telnet、Pingv4 および Pingv6、SNMP などのホスト キープアライブ/ループバック検出 開始 - インターネット キー エクスチェンジ (IKE) プロトコル (IPSec)
WK_CPU_Q_ICMP_GEN(3)	ICMP - 接続先到達不能 ICMP - TTL 期限切れ

CPU キュー (キュー No.)	機能
WK_CPU_Q_ROUTING_CONTROL(4)	

CPU キュー (キュー No.)	機能
	Routing Information Protocol バージョン 1 (RIPv1) RIPv2 Interior Gateway Routing Protocol (IGRP) Border Gateway Protocol (BGP) PIM-UDP 仮想ルータ冗長プロトコル (VRRP) Hot Standby Router Protocol バージョン 1 (HSRPv1) HSRPv2 ゲートウェイ ロード バランシング プロトコル (GLBP) ラベル配布プロトコル (LDP) Web Cache Communication Protocol (WCCP) 次世代 Routing Information Protocol (RIPng) Open Shortest Path First (OSPF) Open Shortest Path First バージョン 3 (OSPFv3) Enhanced Interior Gateway Routing Protocol (EIGRP) Enhanced Interior Gateway Routing Protocol バージョン 6 (EIGRPv6) DHCPv6 プロトコルに依存しないマルチキャスト (PIM) Protocol Independent Multicast バージョン 6 (PIMv6) 次世代 Hot Standby Router Protocol (HSRPng) IPv6 制御 Generic Routing Encapsulation (GRE) キーペアライブ

CPU キュー (キュー No.)	機能
	ネットワークアドレス変換 (NAT) パン ト Intermediate System-to-Intermediate System (IS-IS)
WK_CPU_Q_FORUS_ADDR_RESOLUTION(5)	アドレス解決プロトコル (ARP) IPv6 ネイバーアドバタイズメントおよび ネイバー勧誘
WK_CPU_Q_ICMP_REDIRECT(6)	インターネット制御メッセージプロトコ ル (ICMP) リダイレクト
WK_CPU_Q_INTER_FED_TRAFFIC(7)	内部通信用のレイヤ2ブリッジドメイン 注入。
WK_CPU_Q_L2_LVX_CONT_PACK(8)	Exchange ID (XID) パケット
WK_CPU_Q_EWLC_CONTROL(9)	Embedded Wirelss Controller (eWLC) [ワ イヤレスアクセスポイントの制御とプロ ビジョンング (CAPWAP) (UDP 5246)]
WK_CPU_Q_EWLC_DATA(10)	eWLC データパケット (CAPWAP DATA、UDP 5247)
WK_CPU_Q_L2_LVX_DATA_PACK(11)	不明なユニキャストパケットがマップ要 求のためにパントされました。
WK_CPU_Q_BROADCAST(12)	すべてのタイプのブロードキャスト
WK_CPU_Q_OPENFLOW(13)	学習キャッシュオーバーフロー (レイヤ 2+レイヤ3)
WK_CPU_Q_CONTROLLER_PUNT(14)	データ - アクセスコントロールリスト (ACL) フル データ - IPv4 オプション データ - IPv6 ホップバイホップ データ - リソース不足/すべてをキャッチ データ - リバースパス フォワーディン グ (RPF) が不完全 収集パケット

CPU キュー (キュー No.)	機能
WK_CPU_Q_TOPOLOGY_CONTROL(15)	スパニング ツリー プロトコル (STP) Resilient Ethernet Protocol (REP) Shared Spanning Tree Protocol (SSTP)
WK_CPU_Q_PROTO_SNOOPING(16)	ダイナミック ARP インスペクション (DAI) の Address Resolution Protocol (ARP) スヌーピング
WK_CPU_Q_DHCP_SNOOPING(17)	DHCP スヌーピング
WK_CPU_Q_TRANSIT_TRAFFIC(18)	これは、ソフトウェアパスで処理する必要がある NAT によってパントされたパケットに使用されます。
WK_CPU_Q_RPF_FAILED(19)	データ - mRPF (マルチキャスト RPF) が失敗しました
WK_CPU_Q_MCAST_END_STATION_SERVICE(20)	Internet Group Management Protocol (IGMP) /Multicast Listener Discovery (MLD) 制御
WK_CPU_Q_LOGGING(21)	アクセスコントロールリスト (ACL) ロギング
WK_CPU_Q_PUNT_WEBAUTH(22)	Web 認証
WK_CPU_Q_HIGH_RATE_APP(23)	有線アプリケーションの可視性と制御 (WDAVC) トラフィック ネットワークベースのアプリケーション認識 (NBAR) トラフィック トラフィック分析および分類のための暗号化トラフィック分析 (ETA)
WK_CPU_Q_EXCEPTION(24)	IKE の表示 IP ラーニング違反 IP ポートのセキュリティ違反 IP スタティックアドレス違反 IPv6 スコープチェック リモートコピープロトコル (RCP) 例外 ユニキャスト RPF 失敗

CPU キュー (キュー No.)	機能
WK_CPU_Q_SYSTEM_CRITICAL(25)	メディアシグナリング/ワイヤレスプロキシ ARP
WK_CPU_Q_NFL_SAMPLED_DATA(26)	Netflow サンプルデータと Media Services Proxy (MSP)
WK_CPU_Q_LOW_LATENCY(27)	双方向フォワーディング検出 (BFD)、Precision Time Protocol (PTP)
WK_CPU_Q_EGR_EXCEPTION(28)	出力解決例外
WK_CPU_Q_STACKWISE_VIRTUAL_CONTROL(29)	前面スタッキングプロトコル、つまり SVL
WK_CPU_Q_MCAST_DATA(30)	データ - (S、G) の作成 データ - ローカル結合 データ - PIM 登録 データ - SPT スイッチオーバー データ - マルチキャスト
WK_CPU_Q_GOLD_PKT(31)	Gold

ユーザー設定可能な 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。



- (注) すべての `system-cpp-policy` コンフィギュレーションは、再起動後も保持されるように保存する必要があります。

CPU キューのポリサーの有効化と無効化

CPU キューのポリサ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) を設定します。

CPU キューのポリサ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。



- (注) デフォルトのポリサーがすでに存在する場合は、その削除を慎重に考慮して制御します。そのようにしないと、システムが CPU 占有や制御パケットドロップなどのその他の異常を検出する場合があります。

ポリサーレートの変更

これは、`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レートアクション（パケット/秒単位）を設定することで実行できます。

ポリサーレートを設定する場合、設定したレートは最も近い200の倍数に自動的に変換されることに注意してください。たとえば、CPU キューのポリサーレートを 100 pps に設定すると、システムは 200 に変更します。または、ポリサーレートを 650 に設定すると、システムは 600 に変更します。この動作を示す出力例については、この章の「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」を参照してください。

ポリサーレートをデフォルトに設定

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 キューのポリサーをデフォルト値に設定します。

ソフトウェアバージョンのアップグレードまたはダウングレード

ソフトウェアバージョンのアップグレードと CoPP

デバイスのソフトウェアバージョンをアップグレードすると、システムは CoPP に必要な更新を確認して実行します（たとえば、`system-cpp-policy` ポリシーマップを確認し、欠落している場合は作成します）。また、アップグレードアクティビティの前後に特定のタスクを完了する必要があります。これにより、設定の更新が正しく反映され、CoPP が期待どおりに動作し続けることが保証されます。ソフトウェアのアップグレードに使用する方法に応じて、アップグレード関連のタスクはオプションのシナリオまたは推奨されるシナリオもあれば、必須のシナリオもあります。

ここでは、アップグレードのシステムアクションとユーザーアクションについて説明します。また、リリース固有の警告も含まれます。

アップグレードのシステムアクション

デバイスのソフトウェアバージョンをアップグレードすると、システムは以下のアクションを実行します。これはすべてのアップグレード方法で共通です。

- アップグレード前のデバイスに `system-cpp-policy` ポリシーマップがなかった場合、アップグレード時にシステムはデフォルトポリシーを作成します。
- アップグレード前のデバイスに `system-cpp-policy` ポリシーマップがあった場合、アップグレード時にシステムはポリシーを再生成しません。

アップグレードのユーザーアクション

アップグレードのユーザーアクション（アップグレード方法に応じて）：

アップグレード方法	条件	アクション時間とアクション	目的
標準 ¹	なし	アップグレード後（必須） グローバル コンフィギュレーション モードで cpp system-default コマンドを入力します。	最新のデフォルトのポリサーレートを取得します。

¹ スイッチのリロードを伴うソフトウェアアップグレードの方法を指します。インストールモードまたはバンドルモードにすることができます。

ソフトウェアバージョンのダウングレードと CoPP

ダウングレードのシステムアクションとユーザーアクションについて、ここで説明します。

ダウングレードのシステムアクション

デバイスのソフトウェアバージョンをダウングレードすると、これらのアクションが実行されます。これはすべてのダウングレード方法に適用されます。

- システムは `system-cpp-policy` ポリシーマップをデバイスに保持し、コントロールプレーンにインストールします。

ダウングレードのユーザーアクション

ダウングレードのユーザーアクション：

アップグレード方法	条件	アクション時間とアクション	目的
標準 ²	なし	操作は不要です。	N/A

² スイッチのリロードを伴うソフトウェアアップグレードの方法を指します。インストールモードまたはバンドルモードにすることができます。

ソフトウェアバージョンをダウングレードしてからアップグレードする場合、適用されるシステムアクションとユーザーアクションは、アップグレードについて説明したものと同じです。

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。手順は次のとおりです。

手順の概要

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** *class-name*
5. **police rate** *rate* **pps**
6. **exit**
7. **control-plane**
8. **service-policy input** *policy-name*
9. **end**
10. **show policy-map control-plane**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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： デバイス (config)# policy-map system-cpp-policy デバイス (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 <i>class-name</i> 例：	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

	コマンドまたはアクション	目的
	<pre>デバイス (config-pmap) # class system-cpp-police-protocol-snooping デバイス (config-pmap-c) #</pre>	<p>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。</p>
ステップ 5	<p>police rate rate pps</p> <p>例 :</p> <pre>デバイス (config-pmap-c) # police rate 100 pps デバイス (config-pmap-c-police) #</pre>	<p>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。</p> <p>(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。</p>
ステップ 6	<p>exit</p> <p>例 :</p> <pre>デバイス (config-pmap-c-police) # exit デバイス (config-pmap-c) # exit デバイス (config-pmap) # exit デバイス (config) #</pre>	<p>グローバル コンフィギュレーション モードに戻ります。</p>
ステップ 7	<p>control-plane</p> <p>例 :</p> <pre>デバイス (config) # control-plane デバイス (config-cp) #</pre>	<p>制御プレーン (config-cp) コンフィギュレーションモードを開始します。</p>
ステップ 8	<p>service-policy input policy-name</p> <p>例 :</p> <pre>デバイス (config) # control-plane デバイス (config-cp) # service-policy input system-cpp-policy デバイス (config-cp) #</pre>	<p>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。</p>
ステップ 9	<p>end</p> <p>例 :</p> <pre>デバイス (config-cp) # end</pre>	<p>特権 EXEC モードに戻ります。</p>
ステップ 10	<p>show policy-map control-plane</p> <p>例 :</p> <pre>デバイス # show policy-map control-plane</pre>	<p>system-cpp ポリシーの下で設定されたすべてのクラス、さまざまなトラフィックタイプに設定されたレート、および統計情報を表示します。</p>

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。

手順の概要

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** *class-name*
5. **no police rate** *rate* **pps**
6. **end**
7. **show policy-map control-plane**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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： デバイス (config)# policy-map system-cpp-policy デバイス (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 <i>class-name</i> 例： デバイス (config-pmap)# class system-cpp-police-protocol-snooping デバイス (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。「 <i>CoPP</i> のシステム定義値」の表を参照してください。
ステップ 5	no police rate <i>rate</i> pps 例： デバイス (config-pmap-c)# no police rate 100 pps	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	end 例：	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
	デバイス (config-pmap-c) # end	
ステップ 7	show policy-map control-plane 例 : デバイス # show policy-map control-plane	system-cpp ポリシーの下で設定されたすべてのクラス、およびさまざまなトラフィックタイプと統計情報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。

手順の概要

1. **enable**
2. **configure terminal**
3. **cpp system-default**
4. **end**
5. **show platform hardware fed switch {switch-number} qos que stats internal cpu policer**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例 : デバイス > enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> • パスワードを入力します (要求された場合)。
ステップ 2	configure terminal 例 : デバイス # 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 : デバイス (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。

	コマンドまたはアクション	目的
ステップ 4	end 例： デバイス (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 stats internal cpu policer 例： デバイス # 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 **2000 pps** です。

```

デバイス> enable
デバイス# configure terminal
デバイス (config) # policy-map system-cpp-policy
デバイス (config-pmap) # class system-cpp-police-protocol-snooping
デバイス (config-pmap-c) # police rate 2000 pps
デバイス (config-pmap-c-police) # end
    
```

```

デバイス# show policy-map control-plane
Control Plane
    
```

```

Service-policy input: system-cpp-policy
    
```

```

<output truncated>
    
```

```

Class-map: system-cpp-police-dot1x-auth (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0000 bps, drop rate 0000 bps
  Match: none
  police:
    rate 1000 pps, burst 244 packets
    conformed 0 bytes; actions:
      transmit
    exceeded 0 bytes; actions:
      drop
    
```

```

Class-map: system-cpp-police-protocol-snooping (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0000 bps, drop rate 0000 bps
Match: none
police:
  rate 2000 pps, burst 488 packets
  conformed 0 bytes; actions:
    transmit
  exceeded 0 bytes; actions:
    drop

<output truncated>

Class-map: class-default (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0000 bps, drop rate 0000 bps
Match: any

```

例：CPU キューの無効化

次に、CPU キューをディセーブルにする例を示します。ここでは、**class system-cpp-police-protocol-snooping** CPU キューが無効になります。

```

デバイス> enable
デバイス# configure terminal
デバイス(config)# policy-map system-cpp-policy
デバイス(config-pmap)# class system-cpp-police-protocol-snooping
デバイス(config-pmap-c)# no police rate 100 pps
デバイス(config-pmap-c)# end

```

```

デバイス# 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
  class system-cpp-police-forus
  class system-cpp-default

```

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

<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。



- (注) 一部のCPUキューでは、すべてのクラスにデフォルトレートを設定しても、デフォルトレートと設定レートの値は同じにはなりません。これは、設定レートが最も近い200の倍数に丸められるためです。この動作は、デバイスのクロック速度によって制御されます。下の出力例では、DHCP スヌーピングとNFL SAMPLED DATA のデフォルトレートと設定レートの値にこの違いが示されています。

```

デバイス> enable
デバイス# configure terminal
デバイス(config)# cpp system-default
Defaulting CPP : Policer rate for all classes will be set to their defaults
デバイス(config)# end
    
```

```

Device# show platform hardware fed switch 1 qos queue stats internal cpu policer
CPU Queue Statistics
    
```

QId	PlcIdx	Queue Name	Enabled	(default) Rate	(set) Rate	Queue Drop(Bytes)	Queue Drop(Frames)
0	11	DOT1X Auth	Yes	1000	1000	0	0
1	1	L2 Control	Yes	2000	2000	0	0
2	14	Forus traffic	Yes	4000	4000	0	0
3	0	ICMP GEN	Yes	600	600	0	0
4	2	Routing Control	Yes	5400	5400	0	0
5	14	Forus Address resolution	Yes	4000	4000	0	0
6	0	ICMP Redirect	Yes	600	600	0	0
7	16	Inter FED Traffic	Yes	2000	2000	0	0
8	4	L2 LVX Cont Pack	Yes	1000	1000	0	0
9	16	EWLC Control	Yes	2000	2000	0	0
10	16	EWLC Data	Yes	2000	2000	0	0
11	13	L2 LVX Data Pack	Yes	1000	1000	0	0
12	0	BROADCAST	Yes	600	600	0	0

13	10	Openflow	Yes	100	200	0	0
14	13	Sw forwarding	Yes	1000	1000	0	0
15	8	Topology Control	Yes	13000	13000	0	0
16	12	Proto Snooping	Yes	2000	2000	0	0
17	6	DHCP Snooping	Yes	500	400	0	0
18	9	Transit Traffic	Yes	500	400	0	0
19	10	RPF Failed	Yes	100	200	0	0
20	15	MCAST END STATION	Yes	2000	2000	0	0
21	13	LOGGING	Yes	1000	1000	0	0
22	7	Punt Webauth	Yes	1000	1000	0	0
23	18	High Rate App	Yes	13000	13000	0	0
24	10	Exception	Yes	100	200	0	0
25	3	System Critical	Yes	1000	1000	0	0
26	10	NFL SAMPLED DATA	Yes	100	200	0	0
27	2	Low Latency	Yes	5400	5400	0	0
28	10	EGR Exception	Yes	100	200	0	0
29	5	Stackwise Virtual OOB	Yes	8000	8000	0	0
30	9	MCAST Data	Yes	500	400	0	0
31	10	Gold Pkt	Yes	100	200	0	0

* NOTE: CPU queue policer rates are configured to the closest hardware supported value

CPU Queue Policer Statistics

```

=====
Policer   Policer Accept  Policer Accept  Policer Drop  Policer Drop
Index     Bytes          Frames          Bytes          Frames
-----
0          0              0              0              0
1          0              0              0              0
2          0              0              0              0
3          0              0              0              0
4          0              0              0              0
5          0              0              0              0
6          0              0              0              0
7          0              0              0              0
8          0              0              0              0
9          0              0              0              0
10         0              0              0              0
11         0              0              0              0
12         0              0              0              0
13         0              0              0              0
14         0              0              0              0
15         0              0              0              0
16         0              0              0              0
17         0              0              0              0
    
```

```

18          0          0          0          0
=====
                        CPP Classes to queue map
=====
PlcIdx CPP Class                               : Queues
-----
0      system-cpp-police-data                   : ICMP GEN/BROADCAST/ICMP Redirect/
10     system-cpp-police-sys-data               : Openflow/Exception/EGR Exception/NFL
      SAMPLED DATA/Gold Pkt/RPF Failed/
13     system-cpp-police-sw-forward            : Sw forwarding/LOGGING/L2 LVX Data
Pack/
9      system-cpp-police-multicast             : Transit Traffic/MCAST Data/
15     system-cpp-police-multicast-end-station : MCAST END STATION /
7      system-cpp-police-punt-webauth          : Punt Webauth/
1      system-cpp-police-l2-control            : L2 Control/
2      system-cpp-police-routing-control       : Routing Control/Low Latency/
3      system-cpp-police-system-critical       : System Critical/
4      system-cpp-police-l2lvx-control         : L2 LVX Cont Pack/
8      system-cpp-police-topology-control      : Topology Control/
11     system-cpp-police-dot1x-auth           : DOT1X Auth/
12     system-cpp-police-protocol-snooping    : Proto Snooping/
6      system-cpp-police-dhcp-snooping        : DHCP Snooping/
14     system-cpp-police-forus                 : Forus Address resolution/Forus traffic/
5      system-cpp-police-stackwise-virt-control : Stackwise Virtual OOB/
16     system-cpp-default                     : Inter FED Traffic/EWLC Control/EWLC
Data/
18     system-cpp-police-high-rate-app        : High Rate App/

```

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（ユーザー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、次のコマンドを使用します。

コマンド	目的
show policy-map control-plane	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。
show policy-map system-cpp-policy	system-cpp ポリシーの下で設定されたすべてのクラスとポリサーレートを表示します。
show platform hardware fed switch {switch-number} qos que stats 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。
show platform software fed {switch-number} qos policy target status	ポリシーステータスとターゲットポートタイプに関する情報を表示します。

CoPPの機能履歴

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イ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だ

けを示しています。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シスコ 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。

機能	リリース	機能情報
コントロールプレーンポリシー (CoPP) または CPP	Cisco IOS XE Everest 16.5.1a	この機能が導入されました。 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。 この機能は、CPU キューの有効化および無効化、ポリサーレートの変更、ポリサーレートのデフォルトへの設定、ユーザー定義のクラスマップの作成を行うための CLI 設定オプションを提供します。
CoPP のシステム定義値の変更	Cisco IOS XE Everest 16.6.1	次の新しいシステム定義のクラスが導入されました。 <ul style="list-style-type: none"> • system-cpp-police-stackwise-virt-control • system-cpp-police-l2lvx-control 次の新しい CPU キューが既存の system-cpp-default クラスに追加されました。 <ul style="list-style-type: none"> • WK_CPU_Q_UNUSED (7) • WK_CPU_Q_EWLC_CONTROL(9) • WK_CPU_Q_EWLC_DATA(10) この新しい CPU キューが既存の system-cpp-police-sw-forward に追加されました。WK_CPU_Q_L2_LVX_DATA_PACK (11) この CPU キューは使用できなくなりました。WK_CPU_Q_SGT_CACHE_FULL(27)
設定されているポリサーレートのシステム動作の変更。	Cisco IOS XE Everest 16.6.4	一部の CPU キューでは、すべてのクラスにデフォルトレートを設定しても、デフォルトレートと設定レートの値は同じにはなりません。これは、設定レートが最も近い 200 の倍数に丸められるためです。

機能	リリース	機能情報
<p>ユーザー定義のクラスマップのサポート停止、および CoPP のシステム定義値の変更</p>	<p>Cisco IOS XE Fuji 16.8.1a</p>	<ul style="list-style-type: none"> • このリリース以降、ユーザー定義のクラスマップの作成はサポートされません。 • 新しいシステム定義クラス <code>system-cpp-police-dhcp-snooping</code> が導入されました。 • 新しい CPU キュー <code>WK_CPU_Q_INTER_FED_TRAFFIC</code> が既存の <code>system-cpp-default</code> クラスに追加されました。 • 次の CPU キューは使用できなくなりました。 <ul style="list-style-type: none"> • <code>WK_CPU_Q_SHOW_FORWARD</code> • <code>WK_CPU_Q_UNUSED</code> • 一部の CPU キューのデフォルトポリサーレート (pps) が変更されました。 <ul style="list-style-type: none"> • <code>WK_CPU_Q_EXCEPTION(24)</code> のデフォルトレートが 100 に変更されました。 • <code>system-cpp-default</code> の下のすべての CPU キューのデフォルトレートが 2000 に増えました。 • <code>system-cpp-police-forus</code> の下のすべての CPU キューのデフォルトレートが 4000 に増えました。

機能	リリース	機能情報
CoPPのシステム定義値の変更	Cisco IOS XE Fuji 16.9.1	<p>このリリース以降、18個のシステム定義クラスが <code>system-cpp-policy</code> の下に作成されます。</p> <p>次の新しいシステム定義のクラスが導入されました。</p> <ul style="list-style-type: none"> • <code>system-cpp-police-high-rate-app</code> • <code>system-cpp-police-system-critical</code> <p>これはクラス <code>system-cpp-police-sys-data: CPU queue WK_CPU_Q_OPENFLOW (13)</code> に追加されました。</p> <p>このCPUキューは使用できなくなりました： <code>WK_CPU_Q_LEARNING_CACHE_OVFL(13)</code>。</p>
システム定義のクラスマップの廃止	Cisco IOS XE Fuji 16.9.4	<p>システム定義のクラスマップ <code>system-cpp-police-control-low-priority</code> は廃止されました。</p>

翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。