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，10 ページ
- CoPP のモニタリング，12 ページ
- CoPP に関する追加情報，14 ページ
- コントロール プレーン ポリシングの機能情報，15 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。
- ポリサー レート設定を削除すると、その影響を受けるすべてのキューの CoPP がディセーブルになります。

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-control-low-priority	WK_CPP_POLICE_CONTROL_LOW_PRI(3)	WK_CPU_Q_ICMP_REDIRECT(6) WK_CPU_Q_GENERAL_PUNT(25)	いいえ	500
system-cpp-police-data	WK_CPP_POLICE_DATA(0)	WK_CPU_Q_ICMP_GEN(3) WK_CPU_Q_BROADCAST(12)	Yes	200
system-cpp-default	WK_CPP_POLICE_DEFAULT_POLICER	WK_CPU_Q_DHCP_SNOOPING WK_CPU_Q_SHOW_FORWARD	いいえ	1000
system-cpp-police-dot1x-auth	WK_CPP_POLICE_DOT1X(11)	WK_CPU_Q_DOT1X_AUTH(0)	いいえ	1000
system-cpp-police-forus	WK_CPP_POLICE_FORUS(14)	WK_CPU_Q_FORUS_ADDR_RESOLUTION(5) WK_CPU_Q_FORUS_TRAFFIC(2)	いいえ	1000
system-cpp-police-l2-control	WK_CPP_POLICE_L2_CONTROL(1)	WK_CPU_Q_L2_CONTROL(1)	いいえ	500

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-l2lvx-control				
system-cpp-police-multicast	WK_CPP_POLICE _MULTICAST(9)	WK_CPU_Q_TRANSIT_TRAFFIC(18) WK_CPU_Q_MCAST_DATA(30)	Yes	500
system-cpp-police-multicast-end-station	WK_CPP_POLICE _MULTICAST_SNOOPING(15)	WK_CPU_Q_MCAST_END_ STATION_SERVICE(20)	Yes	200
system-cpp-police-protocol-snooping	WK_CPP_POLICE_PR	WK_CPU_Q_PROTO_ SNOOPING(16)	いいえ	500
system-cpp-police-punt-webauth	WK_CPP_POLICE_ PUNT_WEBAUTH(7)	WK_CPU_Q_ PUNT_WEBAUTH(22)	いいえ	1000
system-cpp-police-routing-control	WK_CPP_POLICE_ ROUTING_ CONTROL(2)	WK_CPU_Q_ROUTING_ CONTROL(4)	yes	500
system-cpp-police-stackwise-virt-control				
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)	Yes	1000

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-sys-data	WK_CPP_POLICE_SYS_DATA(10)	WK_CPU_Q_LEARNING_CACHE_OVFL(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)	Yes	100
system-cpp-police-topology-control	WK_CPP_POLICE_TOPOLOGY_CONTROL(8)	WK_CPU_Q_TOPOLOGY_CONTROL(15)	いいえ	13000

- 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 マップの下のポリサー 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。手順は次のとおりです。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `> enable`	特権 EXEC モードをイネーブルにします。パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configureterminal** 例： `# 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：
	
	`(config)# policy-map system-cpp-policy`
	`(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：
	
	`(config-pmap)# class system-cpp-police-protocol-snooping`
	`(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** *rate* **pps** 例： `(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。

ステップ 6	**service-policy input** *policy-name*
	
	例：
	
	`(config)# control-plane`
	`(config-cp)#service-policy input system-cpp-policy`
	`(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： `(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： `# 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： `> enable`	特権 EXEC モードをイネーブルにします。パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `# 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：
	`(config)# policy-map system-cpp-policy`
	`(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：
	`(config-pmap)# class system-cpp-police-protocol-snooping`
	`(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	**no police rate** *rate* **pps** 例： `(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： `(config-pmap-c)# end`	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
ステップ 7	**show running-config | begin system-cpp-policy** 例： `# 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。

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	**enable** 例： `> enable`	特権 EXEC モードをイネーブルにします。パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `# 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：
	
	`(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： `(config)# end`	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
ステップ 5	**show platform hardware fed active qos queue stats internal cpu policer** 例： `# show platform hardware fed active qos queue stats internal cpu policer`	デバイス固有の内部キュー情報を表示します。

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。

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


# 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。

```
> enable
# configure terminal
(config)# cpp system-default
Defaulting CPP : Policer rate for all classes will be set to their defaults
(config)# end
Device# configure terminal
Device(config)# class-map match-any class-cpp-user
Device(config-cmap)# match dscp cs1
Device(config-cmap)# exit
Device(config)# policy-map system-cpp-policy
Device(config-pmap)# class system-cpp-policy
Device(config-pmap)# exit
Device(config)# policy-map system-cpp-policy
Device(config-pmap)# class class-cpp-user
Device(config-pmap-c)# police rate 2m bps
Device(config-pmap-c)# end
# show platform hardware fed active qos queue stats internal cpu policer

CPU Queue Statistics
============================================================================================
(default) (set) Queue Queue
QId  PlcIdx  Queue     Name               Enabled    Rate  Rate  Drop(Bytes)  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        1000  1000  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        1000  1000  0            0
6    0       ICMP      Redirect           Yes        600   600   0            0
7    16      Unused                       Yes        1000  1000  0            0
8    4       L2        LVX Cont Pack      Yes        1000  1000  0            0
9    16      EWLC      Control            Yes        1000  1000  0            0
10   16      EWLC      Data               Yes        1000  1000  0            0
11   13      L2        LVX Data Pack      Yes        1000  1000  0            0
12   0       BROADCAST                    Yes        600   600   0            0
13   10      Learning  cache ovfl         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   16      DHCP      Snooping           Yes        1000  1000  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   10      Crypto    Control      Yes  100   200   0      0
24   10      Exception        Yes  100   200   0      0
25   3 General   Punt   Yes  200   200   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 Control    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 10364 137 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 4160 65 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

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 :      Learning cache ovfl/Crypto Control/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/
5  system-cpp-police-stackwise-virt-control :    Stackwise Virtual Control/
2  system-cpp-police-routing-control :     Routing Control/Low Latency/
3  system-cpp-police-control-low-priority :      General Punt/
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/
14 system-cpp-police-forus :         Forus Address resolution/Forus traffic/
5  system-cpp-police-stackwise-virt-control :    Stackwise Virtual Control/
16 system-cpp-default :        DHCP Snooping/Unused/EWLC Control/EWLC Data/
Device#
```

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	**enable** 例： > **enable**	特権 EXEC モードをイネーブルにします。パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**show platform software fed active qos policy target status** 例： Device# **show platform software fed active qos policy target status** TCG 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 ?:255 CoPP-Queue-8 0x0000000100000a OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-9 0x0000000100000b OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-10 0x0000000100000c OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-12 0x00000001000019 OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-13 0x0000000100001a OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-14 0x0000000100001b OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-15 0x0000000100001c OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-16 0x0000000100001d OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-17 0x0000000100001e OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-18 0x0000000100001f OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-19 0x00000001000020 OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-20 0x00000001000021 OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-21 0x00000001000022 OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-22 0x00000001000023 OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-23 0x00000001000024 OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-24 0x00000001000025 OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-25 0x00000001000026 OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-26 0x00000001000027 OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-27 0x00000001000028 OUT	ユーザ定義の COPP ポリシーを表示します。 デフォルトにより、デフォルトのポリサー レートで system-cpp-policy が設定されます。次の **show platform software fed active qos policy target status** コマンドの出力例では、32 のユーザ定義のキューがすべて表示されます。

	コマンドまたはアクション	目的
	VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-28 0x00000001000029 OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-29 0x0000000100002a OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-30 0x0000000100002b OUT VALID,SET_INHW system-cpp-policy ?:255 CoPP-Queue-31 0x0000000100002c OUT VALID,SET_INHW system-cpp-policy	

CoPP に関する追加情報

関連資料

関連項目	マニュアル タイトル
MQC QoS コマンド、および CoPP **show** コマンド	『*Command Reference (Catalyst 9400 Series Switches)*』の *IP* マルチキャスト ルーティングに関する章

MIB

MIB	MIB のリンク
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、およびフィーチャ 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

シスコのテクニカル サポート

説明	Link
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。 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、Cisco Notification Service（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。 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。	http://www.cisco.com/support

コントロール プレーン ポリシングの機能情報

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。

表 *2*：コントロール プレーン ポリシングの機能情報

機能名	リリース	機能情報
コントロール プレーン ポリシング	Cisco IOS XE Everest 16.6.1	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、次のプラットフォームに実装されていました。 • Cisco Catalyst 9400 シリーズ スイッチ