Appendix B

Control Plane Policing

Control plane policing (abbreviated as CPP for Cisco IOS routers and as CoPP for Cisco IOS switches) is an application of quality of service (QoS) technologies in a security context that is available on switches and routers running Cisco IOS that allows the configuration of QoS policies that rate limit the traffic handled by the main CPU of the network device. This protects the control plane of the switch from direct denial-of-service (DoS) attacks, reconnaissance activity, and other unexpected flooding of the control plane.

CPP/CoPP protects IOS-based routers and switches by allowing the definition and enforcement of QoS policies that regulate the traffic processed by the main switch CPU (route or switch processor). With CPP/CoPP, these QoS policies are configured to permit, block, or rate limit the packets handled by the main CPU.

A router or switch can be logically divided into four functional components or planes:

- Data plane
- Management plane
- Control plane
- Services plane

The vast majority of traffic travels through the router via the data plane. However, the route/switch processor (which will hereafter be abbreviated as RP, for route processor) must handle certain packets, such as routing updates, keepalives, and network management. This is often referred to as control and management plane traffic.

Because the RP is critical to network operations, any service disruption to it or the control and management planes can result in business-impacting network outages. A DoS attack targeting the RP, which can be perpetrated either inadvertently or maliciously, usually involves high rates of traffic that needs to be punted up to the CPU (from local routing cache or distributed routing engines) that results in excessive CPU utilization on the RP itself. This may be the case with packets destined to nonexistent networks or other

routing failures. This type of attack, which can be devastating to network stability and availability, may display the following symptoms:

- High route processor CPU utilization (near 100 percent).
- Loss of line protocol keepalives and routing protocol updates, leading to route flaps and major network transitions.
- Interactive sessions via the command-line interface (CLI) are slow or completely unresponsive due to high CPU utilization.
- RP resource exhaustion, meaning that resources such as memory and buffers are unavailable for legitimate IP data packets.
- Packet queue backup, which leads to indiscriminate drops (or drops due to lack of buffer resources) of other incoming packets.

CPP/CoPP addresses the need to protect the control and management planes, ensuring routing stability, availability, and packet delivery. It uses a dedicated control plane configuration via the Modular QoS command-line interface (MQC) to provide filtering and rate limiting capabilities for control plane packets.

Packets handled by the main CPU, referred to as control plane traffic, typically include the following:

- Routing protocols.
- Packets destined to the local IP address of the router.
- Packets from network management protocols, such as Simple Network Management Protocol (SNMP).
- Interactive access protocols, such as Secure Shell (SSH) and Telnet.
- Other protocols, such as Internet Control Message Protocol (ICMP), or IP options, might also require handling by the switch CPU.
- Layer 2 packets such as bridge protocol data unit (BPDU), Cisco Discovery Protocol (CDP), DOT1X, and so on.
- Layer 3 protocols such as authentication, authorization, and accounting (AAA), syslog, Network Time Protocol (NTP), Internet Security Association and Key Management Protocol (ISAKMP), Resource Reservation Protocol (RSVP), and so on.

CPP/CoPP leverages the MQC for its QoS policy configuration. MQC allows the classification of traffic into classes and lets you define and apply distinct QoS policies to separately rate limit the traffic in each class. MQC lets you divide the traffic destined to the CPU into multiple classes based on different criteria. For example, four traffic classes could be defined based on relative importance:

- Critical
- Normal

- Undesirable
- Default

After you define the traffic classes, you can define and enforce a QoS policy for each class according to importance. The QoS policies in each class can be configured to permit all packets, drop all packets, or drop only those packets exceeding a specific rate limit.

Note The number of control plane classes is not limited to four, but should be chosen based on local network requirements, security policies, and a thorough analysis of the baseline traffic.

Note It is also important to keep in mind that CoPP/CPP does not protect the switch/router against itself. For example, in some situations SNMP traps or NetFlow exports generated by the device may be excessive and could have detrimental effects on the CPU, the same way a DoS attack might. Therefore, it is beneficial for an administrator to keep this caveat in mind when deploying such management policies.

Defining Control Plane Policing Traffic Classes

Developing a CPP policy starts with the classification of the control plane traffic. To that end, the control plane traffic needs to be first identified and separated into different class maps.

This section presents a classification template that can be used as a model when implementing CPP on IOS routers in addition to when deploying CoPP on Cisco Catalyst switches. This template presents a realistic classification, where traffic is grouped based on its relative importance and protocol type. The template uses eight different classes, which provide a high level of granularity and make it suitable for real-world environments.

Note Even though you can use this template as a reference, the actual number and type of classes needed for a given network can differ and should be selected based on local requirements, security policies, and a thorough analysis of baseline traffic.

This CPP/CoPP template defines these eight traffic classes:

- **Border Gateway Protocol (BGP):** This class defines traffic that is crucial to maintaining neighbor relationships for BGP routing protocol, such as BGP keepalives and routing updates. Maintaining BGP routing protocol is crucial to maintaining con-

nectivity within a network or to an Internet service provider (ISP). Sites that are not running BGP would not use this class.

- **Interior Gateway Protocol (IGP):** This class defines traffic that is crucial to maintaining IGP routing protocols, such as Open Shortest Path First (OSPF), Enhanced Interior Gateway Routing Protocol (EIGRP), and Routing Information Protocol (RIP). Maintaining IGP routing protocols is crucial to maintaining connectivity within a network.
- **Interactive Management:** This class defines interactive traffic that is required for day-to-day network operations. This class would include light volume traffic used for remote network access and management. For example, Telnet, SSH, NTP, SNMP, and Terminal Access Controller Access Control System (TACACS).
- **File Management:** This class defines high-volume traffic used for software image and configuration maintenance. This class would include traffic generated for remote file transfer; for example, Trivial File Transfer Protocol (TFTP) and File Transfer Protocol (FTP).
- **Monitoring/Reporting:** This class defines traffic used for monitoring a router. This kind of traffic should be permitted but should never be allowed to pose a risk to the router. With CPP/CoPP, this traffic can be permitted but limited to a low rate. Examples include packets generated by ICMP echo requests (ping and traceroute) in addition to traffic generated by Cisco IOS IP Service Level Agreements (IP SLAs) to generate ICMP with different differentiated services code point (DSCP) settings to report on response times within different QoS data classes.
- **Critical Applications:** This class defines application traffic that is crucial to a specific network. The protocols that might be included in this class include generic routing encapsulation (GRE), Hot Standby Router Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), Gateway Load Balancing Protocol (GLBP), Session Initiation Protocol (SIP), Data Link Switching (DLSw), Dynamic Host Configuration Protocol (DHCP), Multicast Source Discovery Protocol (MSDP), Internet Group Management Protocol (IGMP), Protocol Independent Multicast (PIM), multicast traffic, and IPsec.
- **Undesirable:** This explicitly identifies unwanted or malicious traffic that should be dropped and denied access to the RP. For example, this class could contain packets from a well-known worm. This class is particularly useful when specific traffic destined to the router should always be denied rather than be placed into a default category. Explicitly denying traffic allows you to collect rough statistics on this traffic using show commands and thereby offers some insight into the rate of denied traffic. Access control list entries (ACEs) used for classifying undesirable traffic may be added and modified as new undesirable applications appear on the network, and therefore you can use these ACEs as a reaction tool.
- **Default:** This class defines all remaining traffic destined to the RP that does not match any other class. MQC provides the default class so that you can specify how

to treat traffic that is not explicitly associated with any other user-defined classes. It is desirable to give such traffic access to the RP, but at a highly reduced rate. With a default classification in place, statistics can be monitored to determine the rate of otherwise unidentified traffic destined to the control plane. After this traffic is identified, further analysis can be performed to classify it. If needed, the other CPP/CoPP policy entries can be updated to account for this traffic.

Deploying Control Plane Policing Policies

Because CPP/CoPP filters traffic, it is critical to gain an adequate level of understanding about the legitimate traffic destined to the RP prior to deployment. CPP/CoPP policies built without proper understanding of the protocols, devices, or required traffic rates involved can block critical traffic, which has the potential of creating a DoS condition. Determining the exact traffic profile needed to build the CPP/CoPP policies might be difficult in some networks.

The following steps follow a conservative methodology that facilitates the process of designing and deploying CPP/CoPP. This methodology uses iterative access control list (ACL) configurations to help identify and to incrementally filter traffic.

To deploy CPP/CoPP, you should perform the six steps that follow.

Step 1: Determine the Classification Scheme for Your Network

Identify the known protocols that access the RP and divide them into categories using the most useful criteria for your specific network. As an example of classification, the eight categories template presented earlier in this section (BGP, IGP, Interactive Management, File Management, Reporting, Critical Applications, Undesirable, and Default) use a combination of relative importance and traffic type. Select a scheme suited to your specific network, which might require a larger or smaller number of classes.

Step 2: Define Classification Access Lists

Configure each ACL to permit all known protocols in its class that require access to the RP. At this point, each ACL entry should have both source and destination addresses set to **any**. In addition, the ACL for the default class should be configured with a single entry, **permit ip any any**. This matches traffic not explicitly permitted by entries in the other ACLs. After the ACLs have been configured, create a class map for each class defined in Step 1, including one for the default class. Then assign each ACL to its corresponding class map.

Note In this step, you should create a separate class map for the default class, instead of using the class default available in some platforms. Creating a separate class map and assigning a **permit ip any any** ACL allows you to identify traffic not yet classified as part of another class.

Each class map should then be associated with a policy map that permits all traffic, regardless of classification. The policy for each class should be set as conform-action transmit exceed-action transmit.

Step 3: Review the Identified Traffic and Adjust the Classification.

Ideally, the classification performed in Step 1 identified all required traffic destined to the router. However, realistically, not all required traffic is identified before deployment, and the **permit ip any any** entry in the default class ACL logs a number of packet matches. Some form of analysis is required to determine the exact nature of the unclassified packets. For example, you can use the **show access-lists** command to see the entries in the ACLs that are in use and to identify any additional traffic sent to the RP. However, to analyze the unclassified traffic, you can use one of these techniques:

- General ACL classification based on traffic characterization and tracing by Cisco routers

Note For more information on this technique, see http://www.cisco.com/en/US/tech/tk59/technologies_tech_note09186a0080149ad6.shtml.

- Packet analyzers

When traffic has been properly identified, adjust the class configuration accordingly. Remove the ACL entries for those protocols that are not used. Add a **permit ip any any** entry for each protocol just identified.

Step 4: Restrict a Macro Range of Source Addresses

Refine the classification ACLs by only allowing the full range of the allocated CIDR block to be permitted as the source address. For example, if the network has been allocated 172.68.0.0/16, permit source addresses from 172.68.0.0/16 where applicable.

This step provides data points for devices or users from outside the CIDR block that might be accessing the equipment. An external BGP (eBGP) peer requires an exception because the permitted source addresses for the session lies outside the CIDR block. This phase might be left on for a few days to collect data for the next phase of narrowing the ACL entries.

Step 5: Narrow the ACL Permit Statements to Authorized Source Addresses

Increasingly limit the source address in the classification ACLs to permit only sources that communicate with the RP. For example, only known network management stations should be permitted to access the SNMP ports on a router.

Step 6: Refine CPP/CoPP Policies by Implementing Rate Limiting

Use the **show policy-map control-plane** command to collect data about the actual policies in place. Analyze the packet count and rate information and develop a rate-limiting policy accordingly. At this point, you might decide to remove the class map and ACL used for the classification of default traffic. If so, you should also replace the previously defined policy for the default class by the class default policy.

Note Table AB-1 shows a set of tested and validated CPP/CoPP rates. Note that the values presented here are solely for illustration purposes, as every environment has different baselines.

Note At the time of this writing, the Catalyst 4500 supports policing rates only in bits per second (bps), but the Catalyst 6500 and Cisco IOS support both bps and packets per second (pps) rate limits. When configuring CPP, rate limiting using a pps rate is generally preferred because it is the per-packet processing that more significantly impacts CPU utilization than the bps rate.

Table B-1 summarizes control plane policing rate examples along with recommended conforming and exceeding actions.

Table B-1 *Control Plane Policing Rate Limits and Actions Examples*

Traffic Class	Rate (pps)	Rate (bps)	Conform Action	Exceed Action
Border Gateway Protocol	500	4,000,000	Transmit	Drop
Interior Gateway Protocol	50	300,000	Transmit	Drop
Interactive Management	100	500,000	Transmit	Drop
File Management	500	6,000,000	Transmit	Drop

Traffic Class	Rate (pps)	Rate (bps)	Conform Action	Exceed Action
Monitoring	125	900,000	Transmit	Drop
Critical Applications	125	900,000	Transmit	Drop
Undesirable	10[1]	32 Kbps[1]	Drop	Drop
Default	100	500,000	Transmit	Drop

[1] The policing rate for the Undesirable CPP class is effectively 0—regardless of whether pps or bps rates are used and also regardless of what values these rates are set to—since both the conforming and exceeding actions for this class are set to drop. However, the policer still performs a metering operation of the rates of these flows which is often useful for management purposes.

Cisco Catalyst 3850 Control Plane Policing

The Catalyst 3850 switch supports control plane policing, but at the time of this writing, this feature is not configurable.

Cisco Catalyst 4500 Control Plane Policing

On Cisco IOS Catalyst switches, CoPP comes into play right after the switching or the routing decision and before traffic is forwarded to the control plane. When CoPP is enabled, the sequence of events (at a high level) is as follows:

1. A packet enters the switch configured with CoPP on the ingress port.
2. The port performs any applicable input port and QoS services.
3. The packet gets forwarded to the switch CPU.
4. The switch CPU makes a routing or a switching decision, determining whether the packet is destined for the control plane.
5. Packets destined for the control plane are processed by CoPP and are dropped or delivered to the control plane according to each traffic class policy. Packets that have other destinations are forwarded normally.

The Catalyst 4500 and Catalyst 6500 series switches implement CoPP similarly. However, CoPP has been enhanced on both platforms to leverage the benefits of their hardware architectures, and as a result each platform provides unique features. Therefore, the CoPP implementations on Catalyst 4500 and Catalyst 6500 series switches are discussed in platform-specific detail in their respective sections within this appendix.

The Catalyst 4500 series switches support CoPP in hardware in a centralized, nondistributed fashion. CoPP policies are centrally configured under the control plane configuration mode and then enforced in hardware by the classification ternary content-address-

able memory (TCAM) and QoS policers of the Supervisor Engine. Figure B-1 shows this CoPP model.

The Catalyst 4500 implementation of CoPP uses MQC to define traffic classification criteria and to specify the configurable policy actions for the classified traffic. MQC

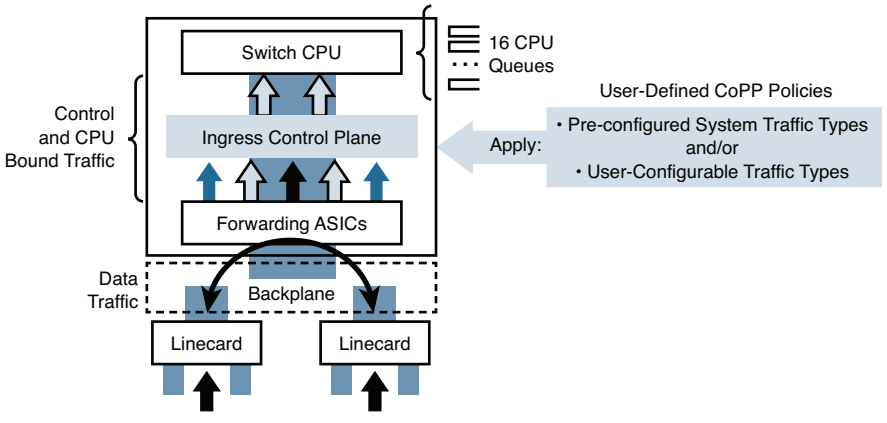

Figure B-1 *Catalyst 4500 Control Plane Policing Implementation*

uses class maps to define packets for a particular traffic class. After you have classified the traffic, you can create policy maps to enforce policy actions for the identified traffic. The control plane global configuration command allows the CoPP service policy to be directly attached to the control plane.

Catalyst 4500 CoPP supports the definition of non-IP traffic classes in addition to IP traffic classes. With this, instead of using the default class for handling all non-IP traffic, you can define separate policies for non-IP traffic. This results in better and more granular control over non-IP protocols, such as Address Resolution Protocol (ARP), Internetwork Packet Exchange (IPX), bridge protocol data units (BPDUs), Cisco Discovery Protocol (CDP), and Secure Socket Tunneling Protocol (SSTP).

One particular characteristic of (Multi-Layer Switch [MLS]-based) Catalyst 4500 CoPP is that the CoPP policy must be named system-cpp-policy. In fact, on these systems, the system-cpp-policy is the only policy map that can be attached to the control plane.

Note This restriction has been removed on MQC-based Catalyst 4500 series switches, beginning with the Supervisor 6-E. However, to maintain backward compatibility, the system-cpp-policy name has been used in this configuration example.

To facilitate the configuration of system-cpp-policy, Catalyst 4500's CoPP provides a global macro function (called system-cpp) that automatically generates and applies CoPP

policies to the control plane. The resulting configuration uses a collection of system-defined class maps for common Layer 3 and Layer 2 control plane traffic. The names of all system-defined CoPP class maps and their matching ACLs contain the prefix system-cpp. By default, no action is specified on any of the system predefined traffic classes. Table B-2 lists the predefined system ACLs.

Table B-2 *Catalyst 4500 System Predefined CoPP ACLs*

Predefined Named ACL	Description
system-cpp-dot1x	MAC DA = 0180.C200.0003
system-cpp-lldp	MAC DA=0180.c200.000E
system-cpp-mcast-cfm	MAC DA=0100.0ccc.ccc0 - 0100.0ccc.ccc7
system-cpp-ucast-cfm	MAC DA=0100.0ccc.ccc0
system-cpp-bpdu-range	MAC DA = 0180.C200.0000 - 0180.C200.000F
system-cpp-cdp	MAC DA = 0100.0CCC.CCCC (UDLD/DTP/VTP/Pagp)
system-cpp-sstp	MAC DA = 0100.0CCC.CCCD
system-cpp-cgmp	MAC DA = 01-00-0C-DD-DD-DD
system-cpp-ospf	IP Protocol = OSPF, IP DA matches 224.0.0.0/24
system-cpp-igmp	IP Protocol = IGMP, IP DA matches 224.0.0.0/3
system-cpp-pim	IP Protocol = PIM, IP DA matches 224.0.0.0/24
system-cpp-all-systems-on-subnet	IP DA = 224.0.0.1
system-cpp-all-routers-on-subnet	IP DA = 224.0.0.2
system-cpp-ripv2	IP DA = 224.0.0.9
system-cpp-ip-mcast-linklocal	IP DA = 224.0.0.0/24
system-cpp-dhcp-cs	IP Protocol = UDP, L4SrcPort = 68, L4DstPort = 67
system-cpp-dhcp-sc	IP Protocol = UDP, L4SrcPort = 67, L4DstPort = 68
system-cpp-dhcp-ss	IP Protocol = UDP, L4SrcPort = 67, L4DstPort = 67

In addition to the predefined classes, you can configure your own class maps matching other control plane traffic. To take effect, these user-defined class maps need to be added to the system-cpp-policy policy map.

CoPP can be deployed on the Catalyst 4500 in one of two main ways:

- You can use the global macro **macro global apply system-cpp** to preconfigure CoPP access lists, class maps, and a system-cpp-policy policy map (with no class actions configured); an administrator can then modify this template and tune it to suit specific environments.

- The CoPP policy can be generated manually (as shown in the Example B-1).

In Example B-1, CoPP has been deployed manually (to keep the policy as consistent as possible between the other CoPP/CPP configuration examples), inline with the previously defined recommendations.

Note It is recommended to include a CPP or CoPP prefix in the ACL and class map names to prevent any potential classification errors for similarly named ACLs and class maps used in data plane policies.

Example B-1 *Catalyst 4500 Control Plane Policing Configuration Example*

```
! This section defines the access lists for the CoPP traffic classes
C4500-E(config)# ip access-list extended COPP-BGP
C4500-E(config-ext-nacl)# remark BGP
C4500-E(config-ext-nacl)# permit tcp host 192.168.1.1 host 10.1.1.1 eq bgp
C4500-E(config-ext-nacl)# permit tcp host 192.168.1.1 eq bgp host 10.1.1.1

C4500-E(config)# ip access-list extended COPP-IGP
C4500-E(config-ext-nacl)# remark IGP (OSPF)
C4500-E(config-ext-nacl)# permit ospf any host 224.0.0.5
C4500-E(config-ext-nacl)# permit ospf any host 224.0.0.6
C4500-E(config-ext-nacl)# permit ospf any any

C4500-E(config)# ip access-list extended COPP-INTERACTIVE-MANAGEMENT
C4500-E(config-ext-nacl)# remark TACACS (return traffic)
C4500-E(config-ext-nacl)# permit tcp host 10.2.1.1 host 10.1.1.1 established
C4500-E(config-ext-nacl)# remark SSH
C4500-E(config-ext-nacl)# permit tcp 10.2.1.0 0.0.0.255 host 10.1.1.1 eq 22
C4500-E(config-ext-nacl)# remark SNMP
C4500-E(config-ext-nacl)# permit udp host 10.2.2.2 host 10.1.1.1 eq snmp
C4500-E(config-ext-nacl)# remark NTP
C4500-E(config-ext-nacl)# permit udp host 10.2.2.3 host 10.1.1.1 eq ntp
C4500-E(config)# ip access-list extended COPP-FILE-MANAGEMENT
C4500-E(config-ext-nacl)# remark (initiated) FTP (active and passive)
C4500-E(config-ext-nacl)# permit tcp 10.2.1.0 0.0.0.255 eq 21 host 10.1.1.1 gt 1023
established
C4500-E(config-ext-nacl)# permit tcp 10.2.1.0 0.0.0.255 eq 20 host 10.1.1.1 gt 1023
C4500-E(config-ext-nacl)# permit tcp 10.2.1.0 0.0.0.255 gt 1023 host 10.1.1.1 gt
1023 established
C4500-E(config-ext-nacl)# remark (initiated) TFTP
C4500-E(config-ext-nacl)# permit udp 10.2.1.0 0.0.0.255 gt 1023 host 10.1.1.1 gt
1023
```

```
C4500-E(config)# ip access-list extended COPP-MONITORING
C4500-E(config-ext-nacl)# remark PING-ECHO
C4500-E(config-ext-nacl)# permit icmp any any echo
C4500-E(config-ext-nacl)# remark PING-ECHO-REPLY
C4500-E(config-ext-nacl)# permit icmp any any echo-reply
C4500-E(config-ext-nacl)# remark TRACEROUTE
C4500-E(config-ext-nacl)# permit icmp any any ttl-exceeded
C4500-E(config-ext-nacl)# permit icmp any any port-unreachable

C4500-E(config)# ip access-list extended COPP-CRITICAL-APPLICATIONS
C4500-E(config-ext-nacl)# remark HSRP
C4500-E(config-ext-nacl)# permit ip any host 224.0.0.2
C4500-E(config-ext-nacl)# remark DHCP
C4500-E(config-ext-nacl)# permit udp host 0.0.0.0 host 255.255.255.255 eq bootps
C4500-E(config-ext-nacl)# permit udp host 10.2.2.8 eq bootps any eq bootps

C4500-E(config)# ip access-list extended COPP-UNDESIRABLE
C4500-E(config-ext-nacl)# remark UNDESIRABLE TRAFFIC
C4500-E(config-ext-nacl)# permit udp any any eq 1434



 ! This section defines the CoPP policy class maps
C4500-E(config)# class-map match-all COPP-BGP
C4500-E(config-cmap)# match access-group name COPP-BGP
 ! Associates COPP-BGP ACL with class map
C4500-E(config)# class-map match-all COPP-IGP
C4500-E(config-cmap)# match access-group name COPP-IGP
 ! Associates COPP-IGP ACL with class map
C4500-E(config)# class-map match-all COPP-INTERACTIVE-MANAGEMENT
C4500-E(config-cmap)# match access-group name COPP-INTERACTIVE-MANAGEMENT
  ! Associates COPP-INTERACTIVE-MANAGEMENT ACL with class map
C4500-E(config)# class-map match-all COPP-FILE-MANAGEMENT
C4500-E(config-cmap)# match access-group name COPP-FILE-MANAGEMENT
  ! Associates COPP-FILE-MANAGEMENT with class map
C4500-E(config)# class-map match-all COPP-MONITORING
C4500-E(config-cmap)# match access-group name COPP-MONITORING
  ! Associates COPP-MONITORING ACL with class map
C4500-E(config)# class-map match-all COPP-CRITICAL-APPLICATIONS
C4500-E(config-cmap)# match access-group name COPP-CRITICAL-APPLICATIONS
  ! Associates COPP-CRITICAL-APPLICATIONS ACL with class map
C4500-E(config)# class-map match-all COPP-UNDESIRABLE
C4500-E(config-cmap)# match access-group name COPP-UNDESIRABLE
  ! Associates COPP-UNDESIRABLE ACL with class map
```

```
 ! This section defines the CoPP policy
C4500-E(config-cmap)# policy-map system-cpp-policy
C4500-E(config-pmap)# class COPP-BGP
C4500-E(config-pmap-c)# police cir 4000000 bc 400000 be 400000
C4500-E(config-pmap-c-police)# conform-action transmit
C4500-E(config-pmap-c-police)# exceed-action drop
 ! Polices BGP to 4 Mbps
C4500-E(config-pmap)# class COPP-IGP
C4500-E(config-pmap-c)# police cir 300000 bc 3000 be 3000
C4500-E(config-pmap-c-police)# conform-action transmit
C4500-E(config-pmap-c-police)# exceed-action drop
 ! Polices IGP to 300 Kbps
C4500-E(config-pmap)# class COPP-INTERACTIVE-MANAGEMENT
C4500-E(config-pmap-c)# police cir 500000 bc 5000 be 5000
C4500-E(config-pmap-c-police)# conform-action transmit
C4500-E(config-pmap-c-police)# exceed-action drop
 ! Polices Interactive Management to 500 Kbps
C4500-E(config-pmap)# class COPP-FILE-MANAGEMENT
C4500-E(config-pmap-c)# police cir 6000000 bc 60000 be 60000
C4500-E(config-pmap-c-police)# conform-action transmit
C4500-E(config-pmap-c-police)# exceed-action drop
 ! Polices File Management to 6 Mbps
C4500-E(config-pmap)# class COPP-MONITORING
C4500-E(config-pmap-c)# police cir 900000 bc 9000 be 9000
C4500-E(config-pmap-c-police)# conform-action transmit
C4500-E(config-pmap-c-police)# exceed-action drop
 ! Polices Monitoring to 900 Kbps
C4500-E(config-pmap)# class COPP-CRITICAL-APPLICATIONS
C4500-E(config-pmap-c)# police cir 900000 bc 9000 be 9000
C4500-E(config-pmap-c-police)# conform-action transmit
C4500-E(config-pmap-c-police)# exceed-action drop
 ! Polices Critical Applications to 900 Kbps
C4500-E(config-pmap)# class COPP-UNDESIRABLE
C4500-E(config-pmap-c)# police cir 32000 bc 3000 be 3000
C4500-E(config-pmap-c-police)# conform-action drop
C4500-E(config-pmap-c-police)# exceed-action drop
 ! Polices all Undesirable traffic (conform action is drop)
C4500-E(config-pmap)# class class-default
C4500-E(config-pmap-c)# police cir 500000 bc 5000 be 5000
C4500-E(config-pmap-c-police)# conform-action transmit
C4500-E(config-pmap-c-police)# exceed-action drop
 ! Polices all other Control Plane traffic to 500  Kbps
```

```
 ! This section attaches the CoPP policy to the control plane
C4500-E(config)#control-plane
C4500-E(config-cp)# service-policy input system-cpp-policy
! Attaches CoPP policy to control plane
```

You can verify the configuration in Example B-1 with the following commands:

- **show class-map**
- **show policy-map**
- **show policy-map control-plane**

Cisco Catalyst 6500 Control Plane Policing

As previously stated, the Catalyst 4500 and Catalyst 6500 series switches implement CoPP similarly. However, CoPP has been enhanced on both platforms to leverage the benefits of their hardware architectures, and as a result each platform provides unique features.

In the Catalyst 6500 series switches, CoPP takes advantage of the processing power present on linecards by implementing a distributed CoPP model. In this platform, the class QoS policies are centrally configured under the control plane configuration mode. When configured, these policies are first applied at the route processor (Multilayer Switch Feature Card [MSFC]) level, and then they get automatically pushed to the Policy Feature Card (PFC) and each Distributed Forwarding Card (DFC). Figure B-2 illustrates this CoPP model.

CoPP is enabled by default on the Catalyst 6500. These default settings provide adequate protection to the control plane in most deployment scenarios. To create manual CoPP policies, the default CoPP policy needs to be disabled. To disable the default CoPP configuration, enter the **no service-policy input policy-default-autocopp** control plane configuration mode command.

Example B-2 shows the corresponding CoPP configuration for the Catalyst 6500 series switches, based on the recommendations previously defined.

Note As previously mentioned, Cisco 6500 CoPP allows the ability to configure rate limits based on either bits per second or packets per second. When configuring CPP, rate limiting using a pps rate is preferred because it is the per-packet processing that more significantly impacts CPU utilization than the bps rate. The pps rates in the following example are based on Table B-1.

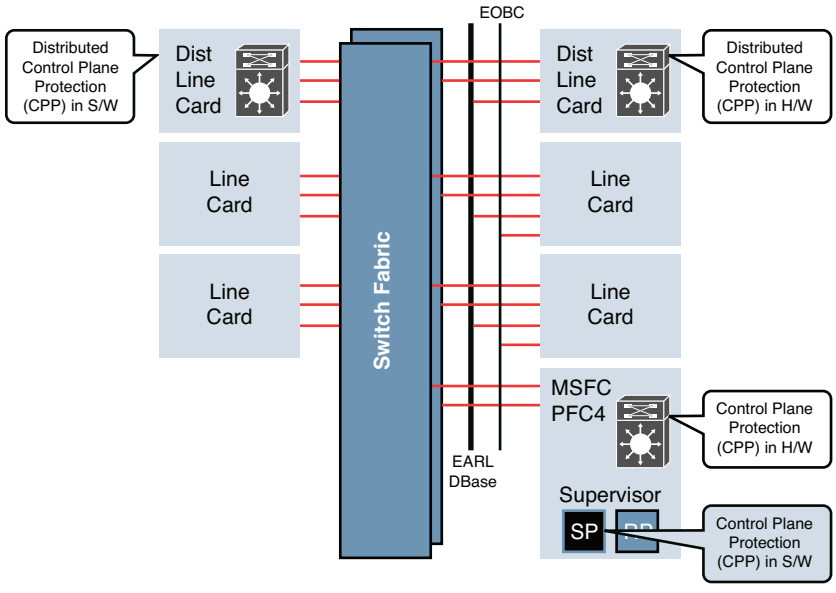

Figure B-2 *Catalyst 6500 Control Plane Policing Implementation*

Note To minimize redundancy, ACLs and class maps—which are identical to the previous example—are not repeated here.

Example B-2 *Catalyst 6500 Control Plane Policing Configuration Example*

```
! This section defines the CoPP policy
C6500-E(config)# policy-map COPP-POLICY
C6500-E(config-pmap)# class COPP-ACL-BGP
C6500-E(config-pmap-c)# police rate 500 pps burst 50
C6500-E(config-pmap-c-police)# conform-action transmit
C6500-E(config-pmap-c-police)# exceed-action drop
 ! Polices BGP to 4 Mbps
C6500-E(config-pmap)# class COPP-ACL-IGP
C6500-E(config-pmap-c)# police rate 50 pps burst 5
C6500-E(config-pmap-c-police)# conform-action transmit
C6500-E(config-pmap-c-police)# exceed-action drop
 ! Polices IGP to 300 Kbps
C6500-E(config-pmap)# class COPP-ACL-INTERACTIVE-MANAGEMENT
C6500-E(config-pmap-c)# police rate 100 pps burst 10
C6500-E(config-pmap-c-police)# conform-action transmit
C6500-E(config-pmap-c-police)# exceed-action drop
 ! Polices Interactive Management to 500 Kbps
C6500-E(config-pmap)# class COPP-ACL-FILE-MANAGEMENT
```

```
C6500-E(config-pmap-c)# police rate 500 pps burst 50
C6500-E(config-pmap-c-police)# conform-action transmit
C6500-E(config-pmap-c-police)# exceed-action drop
 ! Polices File Management to 6 Mbps
C6500-E(config-pmap)# class COPP-ACL-MONITORING
C6500-E(config-pmap-c)# police rate 125 pps burst 12
C6500-E(config-pmap-c-police)# conform-action transmit
C6500-E(config-pmap-c-police)# exceed-action drop
 ! Polices Monitoring to 900 Kbps
C6500-E(config-pmap)# class COPP-ACL-CRITICAL-APPLICATIONS
C6500-E(config-pmap-c)# police rate 125 pps burst 12
C6500-E(config-pmap-c-police)# conform-action transmit
C6500-E(config-pmap-c-police)# exceed-action drop
 ! Polices Critical Applications to 900 Kbps
C6500-E(config-pmap-)# class COPP-ACL-UNDESIRABLE
C6500-E(config-pmap-c)# police rate 10 pps burst 1
C6500-E(config-pmap-c-police)# conform-action drop
C6500-E(config-pmap-c-police)# exceed-action drop
 ! Polices all Undesirable traffic (conform action is drop)
C6500-E(config-pmap)# class class-default
C6500-E(config-pmap-c)# police rate 100 pps burst 10
C6500-E(config-pmap-c-police)# conform-action transmit
C6500-E(config-pmap-c-police)# exceed-action drop
 ! Polices all other Control Plane traffic to 500 Kbps

 ! This section attaches the CoPP policy to the control plane
C6500-E(config)#control-plane
C6500-E(config-cp)# service-policy input COPP-POLICY
 ! Attaches CoPP policy to control plane
```

You can verify the configuration in Example B-2 with the following commands:

- show class-map
- show policy-map
- show policy-map control-plane

Cisco IOS Control Plane Policing (for ASR and ISR Routers)

In Cisco IOS, CPP is implemented in software and comes into play right after the routing decision and before traffic is forwarded to the control plane, as shown in Figure B-3. In addition, Cisco IOS CPP allows for a CPP policy to be applied in the input/output directions.

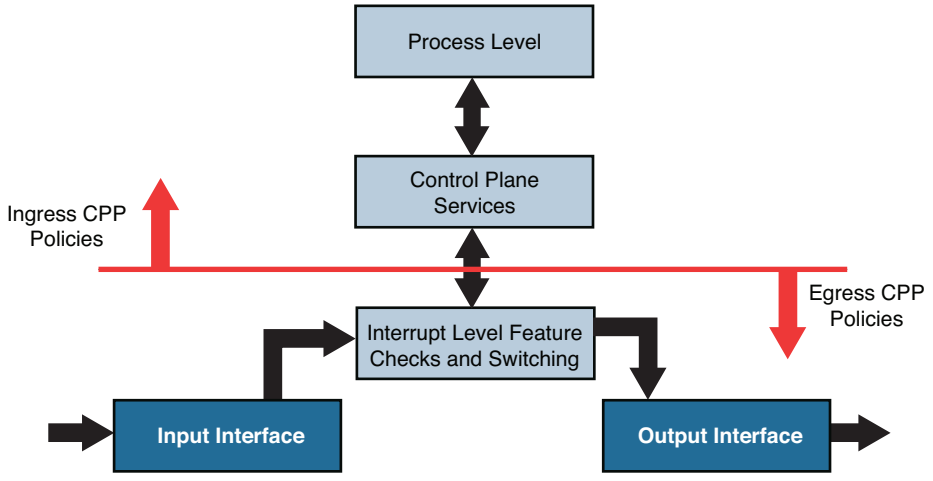

Figure B-3 *Cisco IOS Control Plane Policing Operation*

When CPP is enabled, the sequence of events (at a high level) is as follows:

1. A packet enters the router (configured with CPP) on the ingress interface.
2. The interface performs any applicable input port and QoS services.
3. The packet gets forwarded to the router CPU.
4. The CPU makes a routing or switching decision, determining whether the packet is destined for the control plane.
5. Packets destined for the control plane are processed by CPP and are dropped or delivered to the control plane according to each traffic class policy. Packets that have other destinations are forwarded normally.
6. (Optional) Packets originating from the control plane may also be subject to outbound CPP policies, to prevent the router from being utilized as a source of network reconnaissance or DoS attack.

The corresponding CPP configuration for the Cisco IOS routers, based on the recommendations previously defined, is shown in Example B-3.

Note As previously mentioned, Cisco IOS CPP allows the ability to configure rate limits based on either bits per second or packets per second. When configuring CPP, rate limiting using a pps rate is preferred because it is the per-packet processing that more significantly impacts CPU utilization than the bps rate. The pps rates in the following example are based on Table B-1.

Note To minimize redundancy, ACLs and class maps—which are nearly identical to those in Example B-1—are not repeated here. The only difference is that the prefix CPP is used in ACL and class map and policy map names in this example, as compared to CoPP in the previous examples.

Example B-3 *Cisco IOS Control Plane Policing Configuration Example*

```
! This section defines the CPP-Policy policy map
Router(config)# policy-map CPP-POLICY
Router(config-pmap)# class CPP-ACL-BGP
Router(config-pmap-c)# police rate 500 pps
Router(config-pmap-c-police)# conform-action transmit
Router(config-pmap-c-police)# exceed-action drop
 ! Polices BGP control plane traffic to 500 pps
Router(config-pmap)# class CPP-ACL-IGP
Router(config-pmap-c)# police rate 50 pps
Router(config-pmap-c-police)# conform-action transmit
Router(config-pmap-c-police)# exceed-action drop
 ! Polices IGP control plane traffic to 50 pps
Router(config-pmap)# class CPP-ACL-INTERACTIVE-MANAGEMENT
Router(config-pmap-c)# police rate 100 pps
Router(config-pmap-c-police)# conform-action transmit
Router(config-pmap-c-police)# exceed-action drop
 ! Polices Management control plane traffic to 100 pps
Router(config-pmap)# class CPP-ACL-FILE-MANAGEMENT
Router(config-pmap-c)# police rate 500 pps
Router(config-pmap-c-police)# conform-action transmit
Router(config-pmap-c-police)# exceed-action drop
 ! Polices File-Mgmt control plane traffic to 500 pps
Router(config-pmap)# class CPP-ACL-MONITORING
Router(config-pmap-c)# police rate 125 pps
Router(config-pmap-c-police)# conform-action transmit
Router(config-pmap-c-police)# exceed-action drop
 ! Polices Monitoring control plane traffic to 125 pps
Router(config-pmap)# class CPP-ACL-CRITICAL-APPLICATIONS
Router(config-pmap-c)# police rate 125 pps
Router(config-pmap-c-police)# conform-action transmit
Router(config-pmap-c-police)# exceed-action drop
 ! Polices Critical Applications control plane traffic to 125 pps
Router(config-pmap)# class CPP-ACL-UNDESIRABLE
Router(config-pmap-c)# police rate 10 pps
Router(config-pmap-c-police)# conform-action drop
Router(config-pmap-c-police)# exceed-action drop
```

```
 ! Polices Undesirable control plane traffic to (effectively) 0 pps
 ! As both the conform and exceed action are set to drop
Router(config-pmap)# class class-default
Router(config-pmap-c)# police rate 100 pps
Router(config-pmap-c-police)# conform-action transmit
Router(config-pmap-c-police)# exceed-action drop
 ! Polices all other control plane traffic to 100 pps

! This section applies the CPP policy to the control plane
! The CPP policy is applied in both directions
Router(config)# control-plane
Router(config-cp)# service-policy input CPP-POLICY
 ! Attaches the CPP-POLICY to the control plane in the input direction
Router(config-cp)# service-policy output CPP-POLICY
 ! Attaches the CPP-POLICY to the control plane in the output direction
```

You can verify the configuration in Example B-3 with the following commands:

- **show class-map**
- **show policy-map**
- **show policy-map control-plane** { **input** | **output** }

Additional Reading

Cisco Enterprise Medianet Campus QoS Design 4.0: http://www.cisco.com/en/US/docs/solutions/Enterprise/WAN_and_MAN/QoS_SRND_40/QoSCampus_40.html

Cisco Enterprise Medianet WAN QoS Design 4.0: http://www.cisco.com/en/US/docs/solutions/Enterprise/WAN_and_MAN/QoS_SRND_40/QoSWAN_40.html

Cisco White Paper: Deploying Control Plane Policing: http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6642/prod_white_paper0900aecd-804fa16a.html

Characterizing and Tracing Packet Floods Using Cisco Routers: http://www.cisco.com/en/US/tech/tk59/technologies_tech_note09186a0080149ad6.shtml

Cisco Catalyst 4500 Control Plane Policing Configuration Guide: http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.2/15.02SG/configuration/guide/cntl_pln.html

Cisco Catalyst 6500 Control Plane Policing Configuration Guide: http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/15.0SY/configuration/guide/control_plane_policing_copp.html

Cisco IOS Control Plane Policing Configuration Guide: http://www.cisco.com/en/US/docs/ios-xml/ios/qos_plcshp/configuration/15-0m/qos-plcshp-ctrl-pln-plc.html