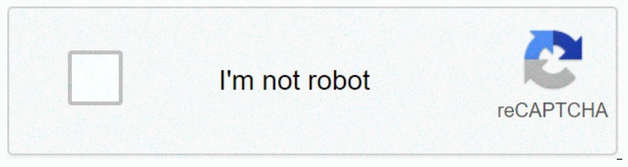

Continue

Cisco asr 9000 qos configuration guide

Got an account? Personalized content Your product and support sign in Forgot your user ID and/or password? Manage Account Page 2 From Release 6.1.2 onwards, Cisco introduces support for 64-bit Linux-based IOS XR operating system. Broad feature parity is maintained between 32-bit and 64-bit environments. Unless explicitly marked otherwise, the content of this document applies to both environments. For more details on the 64-bit Cisco IOS XR, see Release Notes for cisco ASR 9000 Series Routers, Release 6.1.2 document. This guide describes cisco iOS XR QoS configuration. The preface to the Modular QoS Configuration Guide for Cisco ASR 9000 Series Routers contains these sections: This table lists the changes made to this document since it was first published. March 2017 Summary The initial release of this document. July 2017 Republished for Release 6.2.2. Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to Cisco's bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed flawed information about your products and software. Table 1. QoS Features Added or Modified in iOS XR 6.2.x Release Feature Description Modified in Release Where Multi QoS Policy Support Documented This feature allows users to apply aggregate actions to different traffic classes and implement multiple QoS policies on the interface. Release 6.2.1 QoS Multiple Policy Support Overview This example shows a parent and child policy in which two classes are specified in a child's policy. In AF1 classes, the exceed action is set to an action other than to drop traffic. If the child's known order is not configured in the parent policy, the parent police will drop traffic corresponding to the child police suitability level but exceed the parent police suitability level. When used in a parent police, the child's realized order prevents the parent police from dropping incoming traffic in accordance with the level of commitment specified in the child police. In this example, ef classes in child policy are configured with a commitment level of 1 Mbps, appropriate actions and actions exceeded. Traffic under 1 Mbps is presented to the main cop with the MPLS EXP bit set to 4, and traffic exceeding 1 Mbps is dropped. Af1 Classes in child policy are configured with a commitment level of 1 Mbps, appropriate actions and exceeded actions. Traffic under 1 Mbps is presented to the main cop with the MPLS EXP bit set to 3, and traffic exceeding 1 Mbps is presented to the parent police with the MPLS EXP bit set to 2. With this child policy configuration, the parent police sees traffic from the child's class exceed the level 2 Mbps. Without child-conform-aware command in the parent cop, the parent cop becomes 2 Mbps, which can result in dropping some appropriate traffic from the EF class in the child policy. When a child-conform-aware order is configured in the parent police, the parent police does not drop any traffic in accordance with the child's policy. policies-map parent class-default services-child policies level 2 mbps child-conform-aware conformity-action transmit exceed-action drop policy-map ef grade children police level 1 mbps corresponding-action set experimental mpls imposition 4 exceed-action drop class AF1 police level 1 mbps conform-action set experimental mpls imposition 3 exceeds the action set experimental mpls im Load balancing function is a forwarding mechanism for distributing traffic via multiple links based on Layer 3 routing information on the router. There are two types of load balancing schemes: Per-Destination Load Balancing Per-Packet Load Balancing When traffic flow arrives on the router, load balancing per package allows traffic to be evenly distributed between multiple links of the same cost. Schemes per package make routing decisions based on round-robin techniques, regardless of each host's source destination. Only Per-Destination Load Balancing is supported. Load balancing per destination allows the router to distribute packages on top of any of the links in the bundle to achieve load sharing. This scheme is realized through hash calculation based on the source destination address and user session. When load-per-goal balancing is enabled, all packages for a specific source destination pair will go through the same link, although there are several links available. In other words, load balancing per destination can ensure that packages for specific source destination pairs can arrive on an order. By default, load balancing on Layer 2 link bundles is done based on the MAC SA/DA field in the package header. Layer 3 load balancing for link bundles is done on Ethernet Flow Points (EFP) and is based on the source address and IPv4 destination in the package. When Layer 3 service-specific load balancing is configured, all outgoing bundles are balanced based on IPv4 source address and destination. When the package does not have an IPv4 address, the default load balancing is used. Briefly, configuring Multiple QoS policy support involves the following steps—To summarize, two policies (classification and queue policy) are applied towards Egress. Classification policies run traffic first and classify traffic at different priority levels and mark traffic class fields. The queue policy is run second, suitable in the traffic class field to select queues. For matching traffic in classification is preferred to share the same queue, marking the traffic class field with the same value. The show qos interface interface-name output command displays: per class per output policy QoS configuration values queuing policy followed by the classification policy traffic-classes matched by each class in queuing-policy Router#show qos interface TenGigE 0/0/0/3/0 output Interface: TenGigE0/0/0/3/0 output Bandwidth configured: 50000 kbps Bandwidth programed: 50000 kbps ANCP user configured: 0 kbps ANCP programed in HW: 0 kbps Port Shaper programed in HW: 50000 kbps Policy: queue-parent Total number of classes: 4 ---------- Level: 0 Policy: queue-parent Class: class-default Matches: traffic-classes : { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38 , 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63,} and no Traffic class queue: N/A Shape CIR : NONE Shape PIR Profile : 8 (Grid) Scale: 134 PIR: 49920 kbp PBS: 62400 WFQ Profile bytes: 3/9 Committed Weight: 10 Overweight: 10 Bandwidth: 0 kbps, BW sum for Level 0: 0 kbps, Excess Ratio: 1 level ----------: 1 Policy: queue-class children: traffic class-1 Match: traffic class : { 1 } Parent Policy: Parent-queue class: default queue classes: 1040402 (Normal Priority) Queue Limit: 66 kbytes Abs-Index: 19 Template: 0 Curve: 0 Form CIR Profile: WFQ Profile INVALID: 3/19 Committed Weight: 20 Overweight: 20 Bandwidth: 0 kbps , BW jumlah untuk Level 1: 0 kbps, Kelebihan Rasio : 10 ---------- Level: 1 Policy: queue-child Class: traffic-class-2 Matches: traffic-classes : {2} Parent Policy: queue-parent Class: class-default QueueID: 1040403 (Priority Normal) Queue Limit: 126 kbytes Abs-Index: 29 Template: 0 Curve: 0 Shape CIR Profile: INVALID WFQ Profile: 3/39 Committed Weight: 40 Excess Weight: 40 Bandwidth: 0 kbps, BW sum for Level 1: 0 kbps, Excess Ratio: 20 ---------- Level: 1 Policy: queue-child Class: class-default Matches: traffic-classes : { 0, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50 , 51, 52, 53, 54, 55, 56, 57, 58, 59 , 60, 61, 62, 63,} and no Parent Policy traffic class: queue-parent class: QueueID default class: 1040404 (Normal Priority) Queue Limit: 446 kbytes Abs-Index: 52 Template: 0 Curve: 0 Shape CIR Profile: WFQ Profile INVALID: Weight 3/98 Committed: 139 Excess Weight 139 Bandwidth: 0 kbps, BW sum to Level 1: 0 kbps, Excess Ratio: 70 70 Interface: TenGigE0/0/0/3/0 output bandwidth configured: Programmable 100000 kbps bandwidth output: 10000000 kbps CONFIGURED ANCP USERS: 0 kbps ANCP programmed in HW: 0 kbps Programmable Shaper Port in HW: 0 kbps Policy: classification-policy Total number of classes: 5 ---------- ---------- Level: 0 Policy: Class-policy classification: A1 Set traffic-class : 1 QueueID: 0 (Default Port) Police Profile: 59 (Single) Corresponding: 100000 kbps (1 00 mbps) Burst: 1250000 bytes (0 Default) Child Policer Conform: TX Child Policer Exceed: DROP Child Policer Violate: DROP ---------- ---------- Level: 0 Policy: Class-classification policy: B1 Set traffic class : 2 QueueID: 0 (Default Port) Police Profile: 60 (Single) Fit: 200000 kbps (200 mbps) Burst: 2500000 bytes (0 Default) Child Policer Conform: TX Child Policer Exceed: DROP Child Policer Violate: DROP ---------- Level: 0 Policy: Classification-Class Policy : A2 Set of traffic-class : 1 QueueID: 0 (Default Port) Police Profile: 61 (Single) Conform: 300000kbps (300 mbps) Burst: 3750000 bytes (0 Default) Child Policer Conform: TX Child Policer Exceed: DROP Child Policer Violate: DROP ---------- Level: 0 Policy: Class Policy Classification: B2 Set traffic-class : 2 0 (Default Port) Police Profile: 62 (Single) Corresponding: 400000 kbps (400 mbps) Burst: 5000000 bytes (0 Default) Child Policer Conform: TX Child Policer Exceed: DROP Child Policer Violate: DROP ---------- Level: 0 Policy: Class-classification: class-default Queue: 0 (Port Default) ---------- ----------