QoS Configuration Guide, Addendum

December 2018

Supporting AT&T Vyatta Network Operating System

Table of Contents

Copyright Statement

The training materials and other content provided herein for assistance in training on the Vyatta vRouter may have references to Brocade as the Vyatta vRouter was formerly a Brocade product prior to AT&T's acquisition of Vyatta. Brocade remains a separate company and is not affiliated to AT&T.

About This Guide

This addendum describes features that have been added to QoS functionality on the AT&T Vyatta vRouter (referred to as a virtual router, vRouter, or router in the guide) .

Features	vRouter Release	Rev of Qos Configuration Guide 5600 Addendum	Date
Aggregate policing	1801	1	Dec 2018
WRED queue length increased	1801	1	Dec 2018
Make the policer allow for L2 overhead bytes added	1801	1	Dec 2018

Aggregate policing

The AT&T Vyatta vRouter has added support for aggregate policers. An aggregate policer acts upon the traffic across all classes using the action-group in the policy for the target where it is applied. The target can be a physical interface or a VLAN on an interface.

If you apply the same aggregate policy to two different VLANs, the policy does not aggregate the traffic from both VLANs. Rather, it aggregates the traffic per VLAN for the classes that are included within the policy.

You can use an aggregate policer in various ways, such as:

- To monitor statistics on the target's combined traffic flow of the classes included within the aggregate policer.
- To limit the maximum traffic for different classes included within a port or VLAN.

Components of an Aggregate Policer

An aggregate policer is built on a set of nested groups.

Action-Group

At the [edit policy] layer, you must configure an **action group** as a container:

- A policy action group is a police and mark configuration that can be applied to one or more classes in a QoS policy.
- The same action group creates a single police and mark feature per policy which is shared for all the classes using it.
- It allows a single police and mark feature to aggregate different classification streams

Resource Group

A resource group allows multiple values of a specific type to be grouped together and classified together instead of using multiple classifiers. Using a resource group, a QoS class can now classify several values instead of one per class. These resource groups can also be used with firewall rules and with policy-based routing PBR.

There are two types of groups that can be configured at the [edit resources group] layer:

dscp-group	A resource group made up of multiple DSCP values. If this group is referenced by a policy, a match is based on any DSCP value included in the dscp-group.
protocol-group	A resource group made up of multiple IP protocol values. If this group is referenced by a policy, a match is based on any protocol value included in the protocol-group. Note: If a protocol-group is used along with matching a port in a rule, the group can contain only the values supported for matching ports. These protocols are currently: TCP, UDP, UDP-Lite, DCCP, and SCTP.

Configuring an Aggregate Policer

The following example assumes that you have already created a default policy and customer profiles as described in the AT&T QoS Configuration Guide.

Configuring the Action Group

Components of the Action Group Configuration:

Component	Text or Number
action-group	NMC
policy name	policy1
dscp-group	NMC
protocol-group	NMCPROT
profile	prof1

In the following example, the action group lowers the priority of excess traffic if there is a match for any of the values specified within the contained resource-groups. Note that the action group is applied to two classes.

```
vyatta@vyatta# sh policy
policy {
        action {
                name NMC {
                        mark {
                                dscp cs4
                        }
                        police {
                                bandwidth 1mbit
                                then {
                                        mark {
                                                dscp 16
                                        }
                                }
                        }
                }
        }

        qos {
                name policy1 {
                        shaper {
                                class 1 {
                                        match 1 {
                                                action-group NMC
                                                dscp-group NMC
                                        }
                                        profile prof1
                                }
                                class 2{
                                        match 2 {
                                                action-group NMC
                                                protocol-group NMCPROT
                                        }
                                        profile prof1
                                }
                                default prof1
                                profile prof1 {
                                        bandwidth 8mbit
                                }
                        }
                }
        }
}
```

These are the steps used to configure the example:

Steps	Commands
Create an action group named NMC to police traffic classified as dscp cs4 that exceeds 1 megabit per second and to remark this traffic as dscp value 16 (cs2).	`vyatta@R1# set policy action name NMC mark dscp cs4 police bandwidth 1mbit then mark dscp 16`
Specify the aggregate policy name and apply it to class 1 to match the action group NMC.	`vyatta@R1# set policy qos name policy1 shaper class 1 match 1 action-group NMC`
Specify that policy1 includes the dscp-group NMC	`vyatta@R1# set policy qos name policy1 shaper class 1 match 1 dscp-group NMC`
Specify that policy 1 also applies to class 2 to match the action group NMC.	`vyatta@R1# set policy qos name policy1 shaper class 2 match 2 action-group NMC`
Specify that policy 1 includes the protocol-group NMCPROT.	`vyatta@R1# set policy qos name policy1 shaper class 2 match 2 protocol-group NMCPROT`

Steps	Commands
Specify that policy 1 applies to customer profile prof1	`vyatta@R1# set policy qos name policy1 profile prof1`
Specify that policy 1 sends traffic that does not match class 1 or class 2 to the default class set for prof1.	`vyatta@R1# set policy qos name policy1 shaper default prof1`
Specify the bandwidth for the policy1 on prof1. If the traffic exceeds this bandwidth, it will be dropped by default.	`vyatta@R1# set policy qos name policy1 shaper profile prof 1 bandwidth 8mbit`

Configuring the Resource Groups that are Included in the Action Group

The following portion of the configuration creates the resource groups that are included in action-group NMC.

```
[edit]
vyatta@vyatta# sh resources
group {
        dscp-group NMC {
                dscp cs4
                dscp cs6
        }
        protocol-group NMCPROT {
                protocol ospf
        }
}
```

These are the steps used to create the resource groups:

Steps	Commands
Create a dscp-group NMC to include dscp class cs4.	`vyatta@R1# set resources group dscp-group NMC dscp cs4`
Specify that dscp-group NMC also includes dscp class cs6.	`vyatta@R1# set resources group dscp-group NMC dscp cs6`
Create a protocol-group NMCPROT to include protocol OSPF.	`vyatta@R1# set resources group protocol-group NMCPROT ospf`
Commit the configuration.	`vyatta@R1# commit`

WRED Increased Queue Length

The AT&T Vyatta vRouter supports weighted random early detection (WRED) queues of up to 8192 packets long.

High-speed VLANs often use very large WRED queues (4096 or 8192 packets).

Guidelines for WRED Usage

WRED queues should be configured *only* when multiple TCP streams are present. WRED queues are a technique for preventing global TCP synchronization, which is caused when standard QoS queues start to tail-drop many packets. Gobal TCP synchronization occurs when all the TCP steams reduce their transmission windows drastically at the same time, leading to unused bandwidth.

Do not use WRED queues when there are a mixture of TCP and UDP traffic, because the UDP traffic will consume the bandwidth freed up by dropping the TCP packets.

Configuring WRED Queue Length

To configured WRED queues, you must first configure the following two commands:

Syntax:

```
set policy qos name policy-name shaper traffic-class traffic-class random-detect filter-weight
filter-weight

set policy qos name policy-name shaper traffic-class traffic-class random-detect mark-
probability mark-probability
```

Command Parameters

Parameter	Number or Text
policy-name	Name of the QoS policy
traffic-class	The ID of the traffic class. The number ranges from 0 through 3.
filter-weight	The filter weight number ranges from 1 through 12.
mark-probability	The mark-probability number ranges from 1 through 255.

If these commands are not configured and you attempt to configure WRED queue length, the following errors are displayed.

```
filter-weight not set
mark probability not set
```

Guidelines for WRED Queue Length

Generally, you should set the `max-threshold` to the same value as the parameter of the `set policy qos name` *`policy-name`* `shaper traffic-class` *`traffic-class`* `queue-limit` *`queue-limit`* command.

The `queue limit` *`queue-limit`* range is from 1 through 8192 and must be a power of 2.

Setting the `max-threshold` to a lower value than the `queue-limit` will waste memory, because a certain amount of the queue will remain unused.

There are no new commands associated with the increased queue length for WRED. However, vRouter supports an increase in the upper limit that can be specified in the commands below.

policy qos name *policy-name* shaper traffic-class *traffic- class* random-detect min-threshold *min-threshold*

Defines the minimum number of packets for queue length below which packets are not eligible to be randomly dropped. The min-threshold value must always be less than the max-threshold value.

Syntax:

```
set policy qos name policy-name shaper traffic-class traffic-class random-detect
min-threshold min- threshold
```

Syntax:

```
delete policy qos name policy-name shaper traffic-class traffic-class random-detect
min-threshold  [min-threshold ]
```

Syntax:

```
show policy qos name policy-name shaper traffic-class traffic-class random-detect min-threshold
```

Command Parameters

Parameter	Number or Text
policy-name	Name of the QoS policy
traffic-class	The ID of the traffic class. The number ranges from 0 through 3.
min-threshold	The minimum number of packets for queue length. The number ranges from 1 through 8190

Configuration mode

```
policy {
    qos {
        name policy-name {
            traffic-class traffic-class {
                random-detect {
                    min-threshold min-threshold
                }
            }
        }
    }
}
```

- Use the `set` form of this command to define the minimum threshold level for a QoS traffic class.
- Use the `delete` form of this command to delete the minimum threshold level for a QoS traffic class.
- Use the `show` form of this command to display the minimum threshold level for a QoS traffic class.

policy qos name *policy-name* shaper traffic-class *traffic- class* random-detect max-threshold *max-threshold*

Defines the maximum number of packets that can be in a WRED queue, after which all packets will be dropped (tail-dropped, not randomly dropped).

Syntax:

`set policy qos name` *policy-name* **`shaper traffic-class`** *traffic-class* **`random-detect max-threshold`** *max-threshold*

Command Parameters

Parameter	Value or Text
policy-name	Name of the QoS policy
traffic-class	The ID of the traffic class. The number ranges from 0 through 3.
max-threshold	The maximum number of packets for queue length. The number ranges from 1 through 8191.

Configuration mode

```
policy {
    qos {
        name policy-name {
            traffic-class traffic-class {
                random-detect {
                    max-threshold max-threshold
                }
            }
        }
    }
}
```

- Use the `set` form of this command to define the maximum threshold number for a QoS traffic class.
- Use the `delete` form of this command to delete the maximum threshold number for a QoS traffic class.
- Use the `show` form of this command to display the maximum threshold number for a QoS traffic class.

Troubleshooting a WRED Queue

To determine whether a WRED queue is operating effectively, you should monitor the "Tail-drop" and "RED-drop" counters displayed by various "show queuing " commands or the "monitor queuing" command.

```
show queuing
show queuing <interface>
show queuing class
monitor queuing
```

If during normal operation (*not* within a few seconds of a QoS configuration change) the Tail-drop counter is steadily increasing, this increase indicates that the WRED queue is being over-run. It is not randomly dropping enough packets to cause enough TCP streams to back off to reduce the offered traffic to a low enough bandwidth.

Within a few seconds of a QoS configuration change (or immediately after a fresh set of TCP streams start to flow), you can expect to see a temporary increase in Tail-drops before the WRED algorithm has been able to establish the acceptable flow. The WRED algorithm is based on a weighted moving average, so it takes a few iterations to adjust to new flows).

If these Tail-drops stop after a few seconds--even though their count might be high—and if the RED-drop counter is increasing regularly, then the WRED queue could be considered to be working.

If the Tail-drop counter continues to increase, try the following:

- Make the threshold window wider, by reducing the `min-threshold` or increasing the `max-threshold.`
- Reduce the value of the `mark-probability` parameter (to make it more likely that packets are randomly dropped).
- Increase the value of the `filter-weight` parameter.

You should also monitor the `Qlength` counter displayed by the `show queuing` *`interface`* or `monitor queuing` command to try to find the average queue length associated with a WRED queue.

If the queue is under pressure to start randomly dropping packets, you should expect to see the `RED-drop` counter increasing while the Qlength count lies somewhere between the `min-threshold` and the `max-threshold`.

If the `Qlength` count is always very close to the `min-threshold`, it could be a sign that the `mark-probability` parameter is set too low. Note that the mark-probability parameter is actually the inverse probability. That is the probability that a packet will be randomly dropped is related to 1 / mark-probability. So a `mark-probability` of 1 gives a high drop probability, while a `mark-probability` of 255 gives a low drop probability.

Policer Overhead L2 Allowance

The previous frame overhead command, policy qos name *policy-name* shaper frame-overhead *bytes* did not take L2 overhead into consideration for the policer. Therefore, the following commands have been added:

Syntax:

```
set policy qos name policy-name shaper class class-id match rule-name police frame-overhead <inherit | bytes>

set policy action name "name" police frame-overhead <inherit | bytes>
```

Command Parameters

Parameter	Value or Text
policy-name	Name of the QoS policy
class id	The number of the QoS policy class. The number ranges from 1 through 255.
rule-name	The name of the class-matching rule—the rule that specifies the class that must be matched.
Inherit |*bytes*	Specify an L2 overhead allowance: • Inherit: Uses the same value that is specified by the `set policy qos name policy-name shaper frame-overhead bytes` command. • Specify the number of bytes that you want to use for the L2 allowance. The range is from 0 to 1000.