JUNIPER NETWORKS

Technology Overview

Policer Implementation on MX Series, M120, and M320 Routers

Release
12.3

Published: 2012-11-14

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Technology Overview Policer Implementation on MX Series, M120, and M320 Routers
Release 12.3

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

Introduction 1
Policer Implementation Overview 1
Understanding the Benefits of Policers and Token Bucket Algorithms 5
    Scenario 1: Single TCP Connection 5
    Scenario 2: Multiple TCP Connections 6
Determining Proper Burst Size for Traffic Policers 7

Introduction

This document provides information about policer implementation on the Juniper Networks MX Series 3D Universal Edge Routers, the Juniper Networks M120 Multiservice Edge Router, and the Juniper Networks M320 Multiservice Edge Router. The token bucket algorithm is also used by Juniper Networks T Series Core Routers in the implementation of policers. However, the details of the implementation might be somewhat different on T Series Core Routers. Policers are used to rate limit and affect how traffic is handled in the network. This document describes policers and token bucket algorithms, the benefits of implementing policers into your network, and determining the proper burst size.

Policer Implementation Overview

Traffic policing enables you to control the maximum rate of traffic sent or received on an interface and also to partition network traffic into multiple priority levels, also known as classes of service. A policer defines a set of traffic rate limits and sets consequences for traffic that does not conform to the configured limits. Packets in a traffic flow that do not conform to traffic limits are either discarded or marked with a different forwarding class or packet loss priority (PLP) level.

You can apply a policer to inbound or outbound traffic. Policers applied to inbound traffic help to conserve resources by dropping traffic that does not need to be routed through a network. Policers applied to outbound traffic control the bandwidth used.

The Juniper Networks® Junos® operating system (Junos OS) supports three types of policers:

- *Single-rate two-color policer* — The most common policer. Single-rate means that there is only a single bandwidth and burst rate referenced in the policer. The two colors associated with this policer are red (nonconforming) and green (conforming).
- *Single-rate three-color policer* — Similar to the single-rate two-color policer with the addition of the color yellow. This type also introduces the *committed information rate* (CIR) and a *committed burst rate* (CBR).
- *Two-rate three-color policer* — Builds off of the single-rate three-color policer by adding a second rate tier. *Two-rate* means there is an upper bandwidth limit and associated burst size as well as a *peak information rate* (PIR) and a *peak burst rate* (PBS).

NOTE: **The remainder of this topic covers the single-rate two-color policer. For more information about the other types of policers, see the *Junos OS Traffic Policers Configuration Guide*.**

Junos OS policers use a *token bucket algorithm* to enforce a limit on an average transmit or receive rate of traffic at an interface while allowing bursts of traffic up to a maximum value based on the configured bandwidth limit and configured burst size. The token bucket algorithm offers more flexibility than a *leaky bucket algorithm* in that you can

allow a specified traffic burst before starting to discard packets or apply a penalty such as packet output-queuing priority or packet-drop priority.

There are two types of token bucket algorithms that can be used, depending on the type of policer that is applied to network traffic. Single-rate two-color policers use the *single token bucket algorithm* to measure traffic flow conformance to a two-color policer rate limit. Single-rate three-color policers and two-rate three-color policers both use the *dual token bucket algorithm* to measure traffic flow conformance to a three-color policer rate. The main difference between these two token bucket algorithms is that the single token bucket algorithm allows bursts of traffic for short periods, whereas the dual token bucket algorithm allows more sustained bursts of traffic.

NOTE: **The remainder of this topic discusses the single token bucket algorithm. For more information about the dual token bucket algorithm, see the *Junos OS Traffic Policers Configuration Guide*.**

Following are the main components of the token bucket algorithm:

- The *bucket* represents a rate-limiting function of the policer on the interface input or output traffic.
- Each *token* in the bucket represents a “credit” for some number of *bits*, and tokens in the bucket are “cashed in” for the ability to receive or transmit traffic that conforms to a rate limit configured for the policer.
- The *token arrival rate* is a periodic allocation of tokens into the token bucket that is calculated from the configured bandwidth limit.
- The *token bucket depth* defines the capacity of the bucket in *bytes*. Tokens that are allocated after the bucket reaches capacity are not able to be stored and used.

In the token bucket model, the bucket represents the policing function. Tokens are added to the bucket at a fixed rate, but once the specified depth of the bucket is reached, tokens allocated after cannot be stored and used. Each token represents a “credit” for some number of bits, and the tokens in the bucket are “cashed in” for the ability to transmit or receive traffic at the interface. When sufficient tokens are present in the bucket, a traffic flow continues unrestricted.

- The rate at which tokens are added to the bucket represents the highest average transmit or receive rate in bits per second allowed for a given service level. You specify this highest average traffic rate as the *bandwidth limit* of the policer. If the traffic arrival rate is so high that at some point insufficient tokens are present in the bucket, then the traffic flow is no longer conforming to the traffic limit. During periods of relatively low traffic (traffic that arrives at or departs from the interface at average rates below the token arrival rate), unused tokens accumulate in the bucket.
- The depth of the bucket in bytes controls the amount of back-to-back bursting allowed. You specify this factor as the *burst-size limit* of the policer. This second limit affects the average transmit or receive rate by limiting the number of bytes permitted in a transmission burst for a given interval of time. Bursts exceeding the current burst-size

limit are dropped until there are sufficient tokens available to permit the burst to proceed.

To configure a policer, you need to set two parameters:

- Bandwidth limit configured in bps (using the **bandwidth-limit** statement)
- Burst size configured in bytes (using the **burst-size-limit** statement)

NOTE: For single-rate two-color policers only, you can also specify the bandwidth limit as a percentage of either the physical interface port speed or the configured logical interface shaping rate by using the **bandwidth-percent** *percentage* statement. You cannot configure a policer to use bandwidth percentage for aggregate, tunnel, or software interfaces.

Use the following command to set the policer conditions:

```
user@router# set firewall policer <policer name> if-exceeding ?
Possible completions:
  <[Enter]>            Execute this command
+ apply-groups         Groups from which to inherit configuration data
+ apply-groups-except  Don't inherit configuration data from these groups
  bandwidth-limit     Bandwidth limit (8000..100000000000 bits per second)
  bandwidth-percent    Bandwidth limit in percentage (1..100 percent)
  burst-size-limit    Burst size limit (1500..100000000000 bytes)
  |                    Pipe through a command
```

The bandwidth limit parameter is used to determine the average rate limit applied to the traffic, while the burst-size parameter is used to allow for short periods of traffic bursting (back-to-back traffic at average rates that exceed the configured bandwidth limit). Once you apply a set of policer configuration settings (bandwidth limit and burst size), the configured values are adjusted to hardware programmable values. The conversion adjustment introduced is normally less than 1 percent of the configured bandwidth limit. This adjustment is needed because the software allows you to configure the bandwidth limit and burst size to any value within the specified ranges, but those values must be adjusted to the nearest value that can be programmed in the hardware.

The policer bandwidth limit configuration in the hardware is represented by two values: the *credit update frequency* and the *credit size*. The credit update frequency is used by the hardware to determine how frequently tokens (bits of unused bandwidth) are added to the token bucket. The credit size is based on the number of tokens that can fit in the token bucket. The MX Series, M120, and M320 routers contain a set of credit update frequencies instead of having a single credit update frequency to minimize the adjustment difference from the configured bandwidth limit and to support a wide range of policer bandwidth rates (from 40 Kbps to 40 Gbps). One of the frequencies is used to program the policer (bandwidth limit and burst size) in the hardware.

The burst size is based on the overall traffic load and allows bursts of traffic to exceed the configured bandwidth limit. A policer with a large burst size effectively disables the configured bandwidth limit function, so the burst size must be relative to the configured bandwidth limit. You need to consider the traffic patterns in your network before

determining the burst size. For more information about determining burst size, see "Determining Proper Burst Size for Traffic Policers" on page 7.

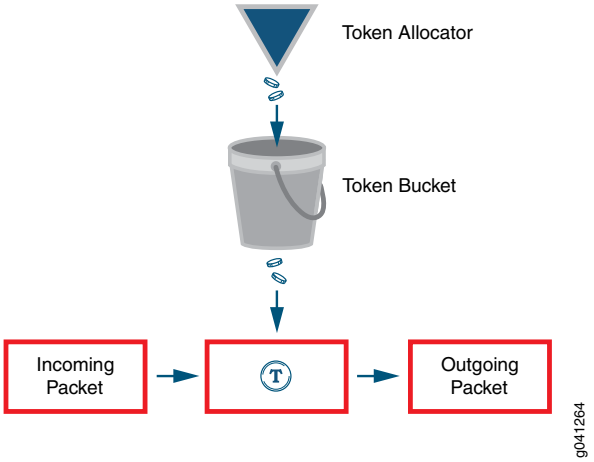
The configured burst size is adjusted in the hardware to a value that is based on the configured bandwidth limit. The burst size extends the configured bandwidth limit for bursty traffic that exceeds the configured bandwidth limit.

When a policer is applied to the traffic at an interface, the initial capacity for traffic bursting is equal to the number of bytes specified in the **burst-size-limit** statement.

Figure 1 on page 4 represents how a policer is implemented using the token bucket algorithm. The token allocator allocates tokens to the policer based on the configured bandwidth limit, which is the token size multiplied by the token arrival rate.

token size x token arrival rate = policer rate (configured bandwidth limit)

Figure 1: Token Bucket Algorithm

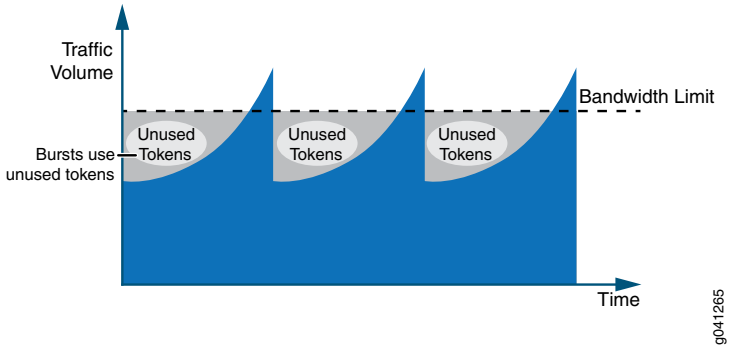


When a packet arrives at an interface configured with a policer, tokens that represent the number of bits that correspond to the length of the packet are used (or "cashed in") from the token bucket. If the token arrival rate is higher than the rate of traffic so that there are tokens not being used, the token bucket is filled to capacity, and arriving tokens "overflow" the bucket and are lost. The token bucket depth represents the single user-configured burst size for the policer.

If there are tokens in the token bucket and the incoming traffic rate is higher than the token rate (the configured policer rate, bandwidth limit), the traffic can use the tokens until the bucket is empty. The token consumption rate can be as high as the incoming traffic rate, which creates the burst of traffic shown in Figure 2 on page 5.

By using the token bucket algorithm, the average bandwidth rate being allowed is close to the configured bandwidth limit while simultaneously supporting bursty traffic, as shown in Figure 2 on page 5.

Figure 2: Traffic Behavior Using Policer and Burst Size



NOTE: The measured length of a packet changes according to the family type that the policer applies to. If the policer is applied under the **family inet** hierarchy, the policer considers only the IPv4 packet length. If the policer is applied under the **family vpls** hierarchy, the entire Ethernet frame (including the Ethernet MAC header) is included in the packet length.

The major factor that affects the policer shaping result is not the conversion adjustment, but the traffic pattern since most network traffic is not consistent and is not sent at a constant rate. Due to the fluctuation of the incoming traffic rate, some of the allocated tokens are not used. As a result, the shaped traffic rate is lower than you might expect, and the TCP connection behavior discussed in "Understanding the Benefits of Policers and Token Bucket Algorithms" on page 5 is a typical example of this. To alleviate this effect of the lower shaped traffic rate, a proper burst size configuration is required.

Related Documentation

- Understanding the Benefits of Policers and Token Bucket Algorithms on page 5
- Determining Proper Burst Size for Traffic Policers on page 7

Understanding the Benefits of Policers and Token Bucket Algorithms

This topic describes some scenarios that demonstrate how difficult it is to control traffic that comes into your network without the help of policers and the token bucket algorithm. These scenarios assume that traffic is coming from a TCP-based connection. Depending on the number of TCP connections, policers can have different affects on rate limits.

This topic presents the following scenarios:

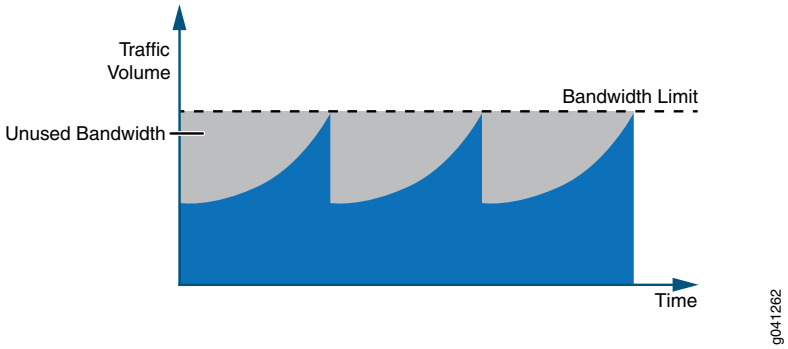
- Scenario 1: Single TCP Connection on page 5
- Scenario 2: Multiple TCP Connections on page 6

Scenario 1: Single TCP Connection

Figure 3 on page 6 shows the traffic loading on an interface with a policer configured. When the traffic rate reaches the configured bandwidth limit (which results in a packet

drop), a TCP slow-start mechanism reduces the traffic rate down to half of what it was. When the traffic rate rises again, the same cycle repeats.

Figure 3: Policer Behavior With a Single TCP Connection

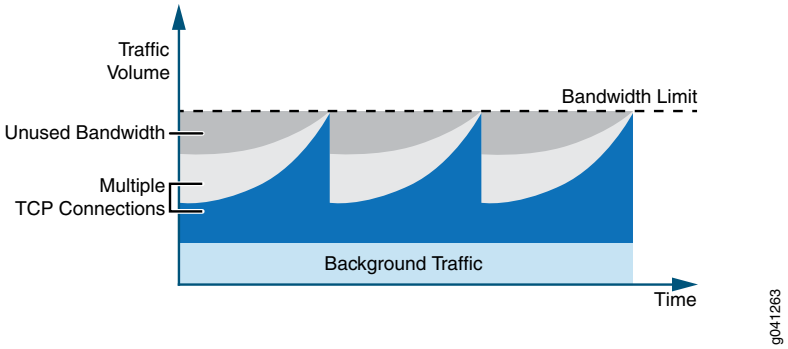


The problem presented in this scenario is that some bandwidth is available, but it is not being used by the traffic. The unused bandwidth shown in Figure 3 on page 6 is the result of an overall data throughput that is lower than the configured bandwidth value. This example is an extreme case because there is only a single TCP connection.

Scenario 2: Multiple TCP Connections

With multiple TCP connections or some background non-TCP-based traffic, there is less unused bandwidth, as depicted in Figure 4 on page 6. However, the same issue of unused bandwidth still exists if all the TCP connections experience a drop when the aggregated traffic rate exceeds the configured bandwidth limit.

Figure 4: Policer Behavior With Background Traffic (Multiple TCP Connections)



To reduce the problem of unused bandwidth in your network, you can configure a burst size.

Related Documentation

- Policer Implementation Overview on page 1
- Determining Proper Burst Size for Traffic Policers on page 7

Determining Proper Burst Size for Traffic Policers

A policer burst-size limit controls the number of bytes of traffic that can pass unrestricted through a policed interface when a burst of traffic pushes the average transmit or receive rate above the configured bandwidth limit. The actual number of bytes of bursty traffic allowed to pass through a policed interface can vary from zero to the configured burst-size limit, depending on the overall traffic load.

By configuring a proper burst size, the effect of a lower shaped rate is alleviated. Use the **burst-size-limit** statement to configure the burst size.

NOTE: **If you set the burst-size limit too low, too many packets will be subjected to rate limiting. If you set the burst-size limit too high, too few packets will be rate-limited.**

Consider these two main factors when determining the burst size to use:

- The allowed duration of a blast of traffic on the line.
- The burst size is large enough to handle the maximum transmission unit (MTU) size of the packets.

The following general guidelines apply to choosing a policer burst-size limit:

- A burst-size limit should not be set lower than 10 times the MTU of the traffic on the interface to be policed.
- The amount of time to allow a burst of traffic at the full line rate of a policed interface should not be lower than 5 ms.
- The minimum and maximum values you can specify for a policer burst-size limit depends on the policer type (two-color or three-color).

BEST PRACTICE: **The preferred method for choosing a burst-size limit is based on the line rate of the interface on which you apply the policer and the amount of time you want to allow a burst of traffic at the full line rate.**

Bursty traffic requires a relatively large burst size so that extra tokens can be allocated into the token bucket for upcoming traffic to use. Figure 5 on page 8 shows an extreme case of bursty traffic where the opportunity to allocate tokens is missed, and the bandwidth goes unused because a large burst size is not configured.

Figure 5: Bursty Traffic Without Configured Burst Size (Excessive Unused Bandwidth)

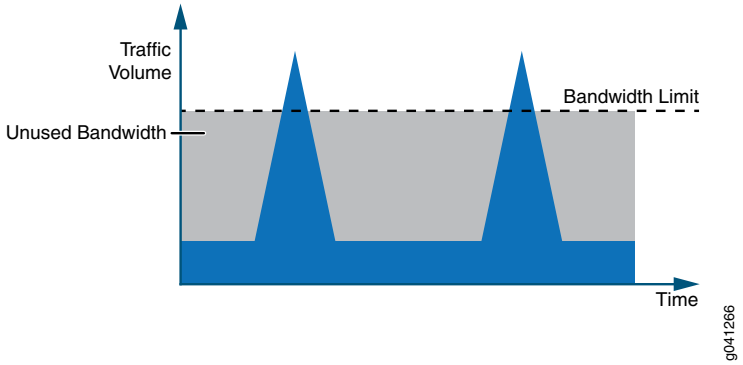
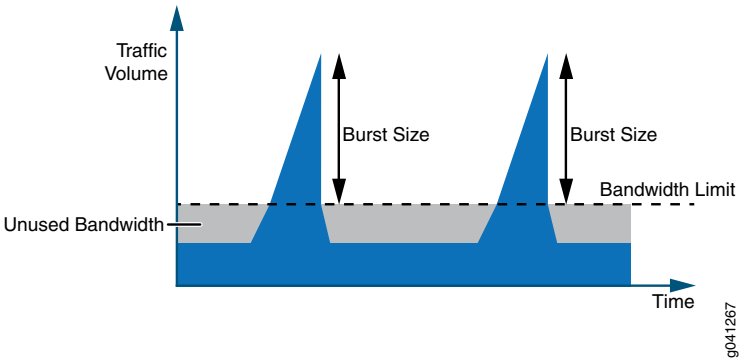


Figure 6 on page 8 depicts how bandwidth usage changes when a large burst size is configured to handle bursty traffic. The large burst size minimizes the amount of unused bandwidth because tokens are being allocated in between the bursts of traffic that can be used during traffic peaks. The burst size determines the depth of the token bucket.

Figure 6: Bursty Traffic With Configured Burst Size (Less Unused Bandwidth)



Configuring a large burst size for the unused tokens creates another issue. If the burst size is set to a very large value, the burst of traffic can be transmitted from the interface at line rate until all the accumulated tokens in the token bucket are used up. This means that configuring a large burst size can allow too many packets to avoid rate limiting, which can lead to a traffic rate that exceeds the bandwidth limit for an extended period of time.

If the average rate is considered within 1 second, the rate is still below the configured bandwidth limit. However, the downstream device might not be able to handle bursty traffic, so some packets might be dropped. As a result, the way to determine the best burst size configuration is to perform experimental configurations, since one burst size is not suitable for every traffic pattern.

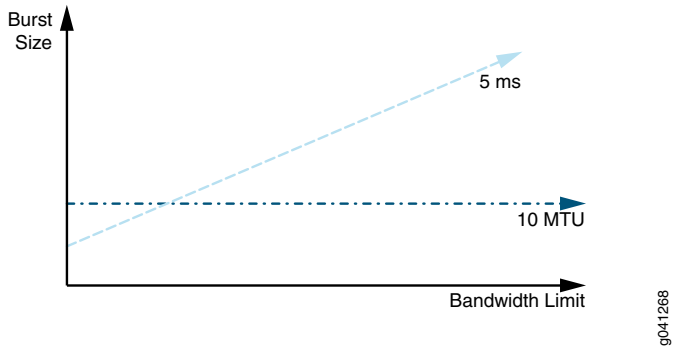
To program the burst size on the MX Series, M120, and M320 routers, the user-configured burst size value is restricted to 1 ms through 600 ms of the policer rate (configured bandwidth limit).

Before performing burst size test configurations, select the initial burst size value using one of the following:

- The recommended formula for calculating burst size for bandwidth described as bits per second is: **burst size = bandwidth x allowable time for burst traffic / 8** (5 ms of the policer rate)
- For policers where the interface bandwidth is unknown, use the MTU method of calculating burst size: **burst size = interface MTU x 10**

Figure 7 on page 9 depicts a comparison between the two methods.

Figure 7: Comparing Burst Size Configuration Methods: 5 ms V.S. 10 MTU



In Figure 7 on page 9, with the 5 ms method, the burst size decreases with the policer rate (the configured bandwidth limit). As a result, the burst size can become so small that it might not be able to hold 2 MTU packets. This can severely affect the policer performance, which makes the 10 MTU method appear to be a better choice.

For example, with a 100 Kbps bandwidth limit configured on a Gigabit Ethernet interface and a burst size configured to 100 ms, the burst size becomes **100 Kbps x 100 ms = 1250 bytes**. This burst size is smaller than one standard MTU payload size on an Ethernet-type interface. A 10 MTU burst size provides a burst size of **1500 bytes x 10 = 15000 bytes**. However, since the maximum burst size is 600 ms of the bandwidth limit, the maximum configured burst size is **100 Kbps x 600 ms = 7500 bytes**. On the Gigabit Ethernet interface, the burst duration is **7500 bytes / 1 Gbps = 60 µs** at Gigabit Ethernet line rate. If the burst size is too large for the downstream device, the burst size can be further reduced until the result is acceptable.

However, if the bandwidth limit is very high, the 10 MTU method might not be able to create a token bucket large enough to accommodate the unused credits. As a result, the average bandwidth limit is lower than what you configured. In this case, the 5 ms method is the better choice for configuring the burst size.

If a 200 Mbps bandwidth limit is configured with a 5 ms burst size, the calculation becomes **200 Mbps x 5 ms = 125 Kbytes**, which is approximately 83 1500-byte packets. If the 200 Mbps bandwidth limit is configured on a Gigabit Ethernet interface, the burst duration is **125000 bytes / 1 Gbps = 1 ms** at the Gigabit Ethernet line rate.

If a large burst size is configured at 600 ms with the bandwidth limit configured at 200 Mbps, the calculation becomes **200 Mbps x 600 ms = 15 Mbytes**. This creates a burst duration of 120 ms at the Gigabit Ethernet line rate. The average bandwidth rate in 1 second becomes **200 Mbps + 15 Mbytes = 320 Mbps**, which is much higher than the configured bandwidth limit at 200 Mbps. This example shows that a larger burst size can affect the measured bandwidth rate.

Related Documentation

- Policer Implementation Overview on page 1
- Understanding the Benefits of Policers and Token Bucket Algorithms on page 5