



Junos[®] OS

Layer 2 Traffic Policing at the Pseudowire

Release

11.1



Published: 2011-02-01

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

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986-1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by the Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, the Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., copyright © 1996, 1997, Maker Communications, Inc.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

Junos® OS Layer 2 Traffic Policing at the Pseudowire

Release 11.1

Copyright © 2011, Juniper Networks, Inc.

All rights reserved. Printed in USA.

Revision History

January 2011—R1 Junos OS 11.1

The information in this document is current as of the date listed in the revision history.

END USER LICENSE AGREEMENT

READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE. BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. **The Parties.** The parties to this Agreement are (i) Juniper Networks, Inc. (if the Customer's principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer's principal office is located outside the Americas) (such applicable entity being referred to herein as "Juniper"), and (ii) the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. **The Software.** In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller, or which was embedded by Juniper in equipment which Customer purchased from Juniper or an authorized Juniper reseller. "Software" also includes updates, upgrades and new releases of such software. "Embedded Software" means Software which Juniper has embedded in or loaded onto the Juniper equipment and any updates, upgrades, additions or replacements which are subsequently embedded in or loaded onto the equipment.

3. **License Grant.** Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

- a. Customer shall use Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.
- b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations are physically contained on a single chassis.
- c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.
- d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.
- e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. **Use Prohibitions.** Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the

Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

5. **Audit.** Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. **Confidentiality.** The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

7. **Ownership.** Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. **Warranty, Limitation of Liability, Disclaimer of Warranty.** The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY LOST PROFITS, LOSS OF DATA, OR COSTS OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, THE SOFTWARE, OR ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. IN NO EVENT SHALL JUNIPER BE LIABLE FOR DAMAGES ARISING FROM UNAUTHORIZED OR IMPROPER USE OF ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. EXCEPT AS EXPRESSLY PROVIDED IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE), INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

9. **Termination.** Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. **Taxes.** All license fees payable under this agreement are exclusive of tax. Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software. If applicable, valid exemption documentation for each taxing jurisdiction shall be provided to Juniper prior to invoicing, and Customer shall promptly notify Juniper if their exemption is revoked or modified. All payments made by Customer shall be net of any applicable withholding tax. Customer will provide reasonable assistance to Juniper in connection with such withholding taxes by promptly: providing Juniper with valid tax receipts and other required documentation showing Customer's payment of any withholding taxes; completing appropriate applications that would reduce the amount of withholding tax to be paid; and notifying and assisting Juniper in any audit or tax proceeding related to transactions hereunder. Customer shall comply with all applicable tax laws and regulations, and Customer will promptly pay or reimburse Juniper for all costs and damages related to any liability incurred by Juniper as a result of Customer's non-compliance or delay with its responsibilities herein. Customer's obligations under this Section shall survive termination or expiration of this Agreement.

11. **Export.** Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to Customer may contain encryption or other capabilities restricting Customer's ability to export the Software without an export license.

12. **Commercial Computer Software.** The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4, FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14 (ALT III) as applicable.

13. **Interface Information.** To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.

14. **Third Party Software.** Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at <http://www.gnu.org/licenses/gpl.html>, and a copy of the LGPL at <http://www.gnu.org/licenses/lgpl.html>.

15. **Miscellaneous.** This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentes confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattache, soient rédigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Table of Contents

Part 1	Overview	
Chapter 1	Layer 2 Traffic Policing at the Pseudowire	3
	Layer 2 Traffic Policing at the Pseudowire Overview	3
Part 2	Configuration	
Chapter 2	Basic Configuration: Configuring Layer 2 Traffic Policing at the Pseudowire	7
	Configuring a Three-Color Layer 2 Policer for the Pseudowire	7
	Configuring a Two-Color Layer 2 Policer for the Pseudowire	8
	Applying the Policers to Dynamic Profile Interfaces	8
	Attaching the Dynamic Profiles to Routing Instances	10
Chapter 3	Complex Configuration: Configuring Layer 2 Traffic Policing at the Pseudowire Using Variables	11
	Using Variables for Layer 2 Traffic Policing at the Pseudowire Overview	11
	Configuring a Policer for the Complex Configuration	12
	Creating a Dynamic Profile for the Complex Configuration	12
	Attaching the Dynamic Profiles to Routing Instances for the Complex Configuration	13
Chapter 4	Configuration Statements for Layer 2 Traffic Policing at the Pseudowire	15
	associate-profile	15
	family vpls (Layer 2 Pseudowires)	16
	layer2-policer	16
	profile-variable-set (Dynamic Profiles)	17
	profile-variable-set (Routing Instances)	17
Part 3	Verification	
Chapter 5	Verifying Your Layer 2 Traffic Policers on VPLS Connections	21
	Verifying Your Layer 2 Traffic Policers on VPLS Connections	21

PART 1

Overview

- Layer 2 Traffic Policing at the Pseudowire on page 3

CHAPTER 1

Layer 2 Traffic Policing at the Pseudowire

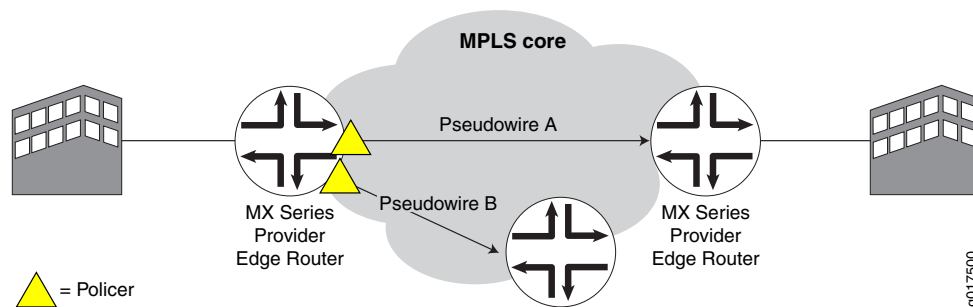
- Layer 2 Traffic Policing at the Pseudowire Overview on page 3

Layer 2 Traffic Policing at the Pseudowire Overview

This feature limits traffic that is sent over the core by policing traffic at the Layer 2 pseudowire level. It uses dynamic profiles to attach two- or three-color policers to pseudowire logical interfaces (ifls). You apply the dynamic profiles to core-facing egress interfaces so that they can police unicast, multicast, and broadcast traffic that is going over the MPLS core.

Figure 1 on page 3 shows an MX Series 3D Universal Edge Router configured as a provider edge (PE) router. It communicates with other PE routers over pseudowires. It can aggregate both unicast and multicast traffic and send it over pseudowires. To limit traffic over the pseudowires, you can set up policers on each pseudowire that faces the MPLS core network.

Figure 1: Limiting Traffic to the Core Using Layer 2 Policers at the Pseudowire Level



NOTE: This feature is supported only on Dense Port Concentrators (DPCs) on MX Series 3D Universal Edge Routers. It is supported only on pseudowire logical interfaces at the egress. It is not supported on tunnel interfaces.

Related Documentation

- Dynamic Profiles for VPLS Pseudowires

PART 2

Configuration

- Basic Configuration: Configuring Layer 2 Traffic Policing at the Pseudowire on page 7
- Complex Configuration: Configuring Layer 2 Traffic Policing at the Pseudowire Using Variables on page 11
- Configuration Statements for Layer 2 Traffic Policing at the Pseudowire on page 15

CHAPTER 2

Basic Configuration: Configuring Layer 2 Traffic Policing at the Pseudowire

- Configuring a Three-Color Layer 2 Policer for the Pseudowire on page 7
- Configuring a Two-Color Layer 2 Policer for the Pseudowire on page 8
- Applying the Policers to Dynamic Profile Interfaces on page 8
- Attaching the Dynamic Profiles to Routing Instances on page 10

Configuring a Three-Color Layer 2 Policer for the Pseudowire

For the basic configuration of Layer 2 policers for pseudowires, create a three-color policer. This scenario shows a two-rate three-color-marking (trTCM) policer.

To configure a three-color policer:

1. Create a three-color policer.

```
[edit firewall]  
user@host# edit three-color-policer trTCM-policer
```

2. Specify that the policer is to be used on a logical interface.

```
[edit firewall three-color-policer trTCM-policer]  
user@host# set logical-interface-policer
```

3. Set the action for the policer.

```
[edit firewall three-color-policer trTCM-policer]  
user@host# set action loss-priority high then discard
```

4. Specify that the policer is a two-rate policer, and configure the policer.

```
[edit firewall three-color-policer trTCM-policer]  
user@host# edit two-rate  
user@host# set color-aware  
user@host# set committed-information-rate 10m  
user@host# set committed-burst-size 50m  
user@host# set committed-burst-size 150k  
user@host# set peak-information-rate 50m  
user@host# set peak-burst-size 450k
```

Related Documentation

- Layer 2 Traffic Policing at the Pseudowire Overview on page 3
- Two-Rate Three-Color Policer Overview
- Configuring a Two-Color Layer 2 Policer for the Pseudowire on page 8
- Applying the Policers to Dynamic Profile Interfaces on page 8
- Attaching the Dynamic Profiles to Routing Instances on page 10

Configuring a Two-Color Layer 2 Policer for the Pseudowire

For the basic configuration of Layer 2 policers for pseudowires, create a two-color policer.

To configure a two-color policer:

1. Create a two-color policer.

```
[edit firewall]  
user@host# edit policer 2color-l2-policer
```

2. Specify that the policer is to be used on a logical interface.

```
[edit firewall policer 2color-l2-policer]  
user@host# set logical-interface-policer
```

3. Configure the policer.

```
[edit firewall policer 2color-l2-policer]  
user@host# edit if-exceeding  
[edit firewall policer 2color-l2-policer if-exceeding]  
user@host# set bandwidth-limit 30m  
user@host# set burst-size-limit 300k
```

4. Set the action that the policer takes to loss-priority, and specify that the packet loss priority (PLP) is high.

```
[edit firewall policer 2color-l2-policer]  
user@host# set then loss-priority high
```

Related Documentation

- Layer 2 Traffic Policing at the Pseudowire Overview on page 3
- Configuring a Three-Color Layer 2 Policer for the Pseudowire on page 7
- Applying the Policers to Dynamic Profile Interfaces on page 8
- Attaching the Dynamic Profiles to Routing Instances on page 10

Applying the Policers to Dynamic Profile Interfaces

This configuration shows how to apply policers to a dynamic profile.

Before you can apply policers, you need to have configured your policers as described in:

- Configuring a Three-Color Layer 2 Policer for the Pseudowire on page 7
- Configuring a Two-Color Layer 2 Policer for the Pseudowire on page 8

To configure the dynamic profiles:

1. Create a dynamic profile for the three-color policer.

```
[edit dynamic-profiles]
user@host# edit pw-trTCM-policer
```

2. Create a dynamic profile interface that has a dynamic underlying interface unit.

```
[edit dynamic-profiles pw-trTCM-policer]
user@host# edit interfaces $junos-interface-ifd-name unit
$junos-underlying-interface-unit
```

3. Specify that VPLS is the protocol family.

```
[edit dynamic-profiles pw-trTCM-policer interfaces "$junos-interface-ifd-name" unit
"$junos-underlying-interface-unit"]
user@host# set family vpls
```

4. Assign the three-color policer to the dynamic profile.

```
[edit dynamic-profiles pw-trTCM-policer interfaces "$junos-interface-ifd-name" unit
"$junos-underlying-interface-unit"]
user@host# set layer2-policer output-three-color trTCM-policer
```

5. Create a dynamic profile for the two-color policer.

```
[edit dynamic-profiles]
user@host# edit pw-2color-l2-policer
```

6. Create a dynamic profile interface that has a dynamic underlying interface unit.

```
[edit dynamic-profiles pw-2color-l2-policer]
user@host# edit interfaces $junos-interface-ifd-name unit
$junos-underlying-interface-unit
```

7. Specify that VPLS is the protocol family.

```
[edit dynamic-profiles pw-2color-l2-policer interfaces "$junos-interface-ifd-name"
unit "$junos-underlying-interface-unit"]
user@host# set family vpls
```

8. Assign the two-color policer to the dynamic profile.

```
[edit dynamic-profiles pw-2color-l2-policer interfaces "$junos-interface-ifd-name"
unit "$junos-underlying-interface-unit"]
user@host# set layer2-policer output-policer 2color-l2-policer
```

Related Documentation

- Layer 2 Traffic Policing at the Pseudowire Overview on page 3
- Configuring a Three-Color Layer 2 Policer for the Pseudowire on page 7
- Configuring a Two-Color Layer 2 Policer for the Pseudowire on page 8
- Attaching the Dynamic Profiles to Routing Instances on page 10

Attaching the Dynamic Profiles to Routing Instances

To bind the dynamic profile to the pseudowire, attach it to a routing instance. The routing instance can be a VPLS instance type or a virtual switch instance type. You can attach dynamic profiles to the routing instance at the VPLS protocol level, at the mesh-group level, or at the neighbor level.

Because this feature is not supported on tunnel interfaces, for VPLS routing interfaces you must include the **no-tunnel-services** statement at the **[edit routing-instances routing-instance-name protocols vpls]** hierarchy level.

- To attach the dynamic profile at the VPLS protocol level:

```
[edit routing-instances]
user@host# edit green protocols vpls associate-profile
[edit routing-instances green protocols vpls associate-profile]
user@host# set pw-2color-l2-policer
```

- To attach the dynamic profile at the mesh-group level.

```
[edit routing-instances]
user@host# edit green protocols vpls mesh-group lata-1 associate-profile
[edit routing-instances green protocols vpls mesh-group lata-1 associate-profile]
user@host# set pw-trTCM-policer
```

- To attach the dynamic profile at the neighbor level.

```
[edit routing-instances]
user@host# edit green protocols vpls mesh-group lata-1 neighbor 10.10.1.1
associate-profile
[edit routing-instances green protocols vpls mesh-group lata-1 neighbor 10.10.1.1
associate-profile]
user@host# set pw-2color-l2-policer
```

Related Documentation

- Layer 2 Traffic Policing at the Pseudowire Overview on page 3
- Configuring a Three-Color Layer 2 Policer for the Pseudowire on page 7
- Configuring a Two-Color Layer 2 Policer for the Pseudowire on page 8
- Applying the Policers to Dynamic Profile Interfaces on page 8

CHAPTER 3

Complex Configuration: Configuring Layer 2 Traffic Policing at the Pseudowire Using Variables

- Using Variables for Layer 2 Traffic Policing at the Pseudowire Overview on page 11
- Configuring a Policer for the Complex Configuration on page 12
- Creating a Dynamic Profile for the Complex Configuration on page 12
- Attaching the Dynamic Profiles to Routing Instances for the Complex Configuration on page 13

Using Variables for Layer 2 Traffic Policing at the Pseudowire Overview

To reduce the number of dynamic profiles needed to police traffic at the core, you can use a variable for the output policer in your dynamic profiles. The variable that you define is called **junos-layer2-output-policer**. The variable is a placeholder that gets filled in when the dynamic profile obtains the value from the routing instance.

To use variables for policers for Layer 2 pseudowires:

1. Create policers.
2. Create a dynamic profile and add a profile variable set to the dynamic profile.
3. In the profile variable set assign a value to the **junos-layer2-output-policer** variable. This value is the name of one of your policers.
4. In the dynamic profile interface configuration at the **[edit dynamic-profiles *profile-name* interfaces "\$junos-interface-ifd-name" unit "\$junos-underlying-interface-unit"]** hierarchy, assign **junos-layer2-output-policer** as one of your Layer 2 policers.
5. When you attach the dynamic profile to a routing instance, assign the profile variable set that you configured in the dynamic profile as the **associate-profile** value.
6. Attach the dynamic profile and the profile variable set to the routing instance.

Related Documentation

- Layer 2 Traffic Policing at the Pseudowire Overview on page 3
- Configuring a Policer for the Complex Configuration on page 12

- [Creating a Dynamic Profile for the Complex Configuration on page 12](#)
- [Attaching the Dynamic Profiles to Routing Instances for the Complex Configuration on page 13](#)

Configuring a Policer for the Complex Configuration

For the complex configuration of Layer 2 policers for pseudowires, create a two-color policer.

To configure a two-color policer:

1. Create a two-color policer.

```
[edit firewall]
user@host# edit policer 10m-policer
```

2. Specify that the policer is to be used on a logical interface.

```
[edit firewall policer 10m-policer]
user@host# set logical-interface-policer
```

3. Configure the policer.

```
[edit firewall policer 10m-policer]
user@host# edit if-exceeding
[edit firewall policer 10m-policer if-exceeding]
user@host# set bandwidth-limit 10m
user@host# set burst-size-limit 100k
```

4. Set the action that the policer takes to loss-priority, and specify that the packet loss priority (PLP) is high.

```
[edit firewall policer 10m-policer]
user@host# set then loss-priority high
```

Related Documentation

- [Layer 2 Traffic Policing at the Pseudowire Overview on page 3](#)
- [Using Variables for Layer 2 Traffic Policing at the Pseudowire Overview on page 11](#)
- [Creating a Dynamic Profile for the Complex Configuration on page 12](#)
- [Attaching the Dynamic Profiles to Routing Instances for the Complex Configuration on page 13](#)

Creating a Dynamic Profile for the Complex Configuration

For this configuration, the dynamic profile defines a profile variable set, and then assigns the variable to the output policer.

To configure the dynamic profiles:

1. Create a dynamic profile.

```
[edit dynamic-profiles]
user@host# edit pw-policer
```

2. Create a profile variable set and define the **junos-layer2-output-policer** variable. In this scenario, set the variable to the **10m-policer**.

```
[edit dynamic-profiles pw-policer]
user@host# edit profile-variable-set pw-policer-var-set
user@host# set junos-layer2-output-policer 10m-policer
```

3. Create a dynamic profile interface that has a dynamic underlying interface unit.

```
[edit dynamic-profiles pw-policer]
user@host# edit interfaces $junos-interface-ifd-name unit
$junos-underlying-interface-unit
```

4. Assign the **junos-layer2-output-policer** variable to the two-color output policer.

```
[edit dynamic-profiles pw-policer interfaces "$junos-interface-ifd-name" unit
"$junos-underlying-interface-unit"]
user@host# set layer2-policer output-policer $junos-layer2-output-policer
```

5. Specify that VPLS is the protocol family.

```
[edit dynamic-profiles pw-2color-l2-policer interfaces "$junos-interface-ifd-name"
unit "$junos-underlying-interface-unit"]
user@host# set family vpls
```

Related Documentation

- Layer 2 Traffic Policing at the Pseudowire Overview on page 3
- Using Variables for Layer 2 Traffic Policing at the Pseudowire Overview on page 11
- Configuring a Policer for the Complex Configuration on page 12
- Attaching the Dynamic Profiles to Routing Instances for the Complex Configuration on page 13

Attaching the Dynamic Profiles to Routing Instances for the Complex Configuration

To bind the dynamic profile to the pseudowire, attach it to a routing instance. When your dynamic profile contains variables, you assign one of the profile variable sets that you configured in your dynamic profile when you associate the profile with the routing instance.

The routing instance can be a VPLS instance type or a virtual switch instance type. You can attach the dynamic profile and the profile variable set to the routing instance at the VPLS protocol level, at the mesh-group level, or at the neighbor level.

Because this feature is not supported on tunnel interfaces, for VPLS routing interfaces you must include the **no-tunnel-services** statement at the **[edit routing-instances routing-instance-name protocols vpls]** hierarchy level.

- To attach the dynamic profile and the profile variable set at the VPLS protocol level:

```
[edit routing-instances]
user@host# edit green protocols vpls associate-profile
[edit routing-instances green protocols vpls associate-profile]
user@host# set profile-variable-set pw-policer
user@host# set profile-variable-set pw-policer-var-set
```

- To attach the dynamic profile and the profile variable set at the mesh-group level:

```
[edit routing-instances]
user@host# edit green protocols vpls mesh-group lata-1 associate-profile
[edit routing-instances green protocols vpls mesh-group lata-1 associate-profile]
user@host# set profile-variable-set pw-policer
user@host# setprofile-variable-set pw-policer-var-set
```

- To attach the dynamic profile and the profile variable set at the neighbor level:

```
[edit routing-instances]
user@host# edit green protocols vpls mesh-group lata-1 neighbor 10.10.1.1
  associate-profile
[edit routing-instances green protocols vpls mesh-group lata-1 neighbor 10.10.1.1
  associate-profile]
user@host# profile-variable-set pw-policer
user@host# profile-variable-set pw-policer-var-set
```

**Related
Documentation**

- [Layer 2 Traffic Policing at the Pseudowire Overview on page 3](#)
- [Using Variables for Layer 2 Traffic Policing at the Pseudowire Overview on page 11](#)
- [Configuring a Policer for the Complex Configuration on page 12](#)
- [Creating a Dynamic Profile for the Complex Configuration on page 12](#)

CHAPTER 4

Configuration Statements for Layer 2 Traffic Policing at the Pseudowire

associate-profile

Syntax	<pre>associate-profile { <i>dynamic-profile-name</i>; profile-variable-set <i>profile-variable-set-name</i>; }</pre>
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i> protocols vpls] [edit routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i>] [edit routing-instances <i>routing-instance-name</i> protocols vpls mesh-group <i>mesh-group-name</i> neighbor neighbor-id]
Release Information	Statement introduced in Junos OS Release 11.1.
Description	Associate a dynamic profile or a profile variable set with a VPLS instance.
Options	<i>dynamic-profile-name</i> —Name of the dynamic profile to attach to this routing instance. The remaining option is explained separately.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Attaching the Dynamic Profiles to Routing Instances on page 10Attaching the Dynamic Profiles to Routing Instances for the Complex Configuration on page 13

family vpls (Layer 2 Pseudowires)

Syntax	family vpls;
Hierarchy Level	[edit dynamic-profiles <i>profile-name</i> interfaces <i>interface-name</i> unit <i>logical-unit-number</i>]
Release Information	Statement introduced in Junos OS Release 11.1.
Description	Specify that the protocol family for the logical interface is VPLS.
Required Privilege Level	router—To view this statement in the configuration. router-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Applying the Policers to Dynamic Profile Interfaces on page 8• Creating a Dynamic Profile for the Complex Configuration on page 12

layer2-policer

Syntax	layer2-policer { output-policer <i>policer-name</i> ; output-three-color <i>policer-name</i> ; }
Hierarchy Level	[edit dynamic-profiles <i>profile-name</i> interfaces <i>logical- interface-name-variable</i>]
Release Information	Statement introduced in Junos OS Release 11.1.
Description	Specify the output policers to be used in the dynamic profile.
Options	<p>output-policer <i>policer-name</i>—Two-color output policer to associate with the interface. You define this policer at the [edit firewall policer] hierarchy level.</p> <p>output-three-color <i>policer-name</i>—Tricolor output policer to associate with the interface. You define this policer at the [edit firewall] hierarchy level.</p>
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Applying the Policers to Dynamic Profile Interfaces on page 8• Creating a Dynamic Profile for the Complex Configuration on page 12

profile-variable-set (Dynamic Profiles)

Syntax	<pre>profile-variable-set { variable-set-name { junos-layer2-output-policer <i>policer-name</i>; } }</pre>
Hierarchy Level	[edit dynamic-profiles <i>profile-name</i>]
Release Information	Statement introduced in Junos OS Release 11.1.
Description	Specify the policer used in the variable set.
Options	junos-layer2-output-policer <i>policer-name</i> —Layer 2 policer that you want to substitute in the dynamic profile. You define this policer at the [edit firewall policer] hierarchy level.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Applying the Policers to Dynamic Profile Interfaces on page 8 Creating a Dynamic Profile for the Complex Configuration on page 12

profile-variable-set (Routing Instances)

Syntax	<pre>profile-variable-set <i>variable-set-name</i></pre>
Hierarchy Level	[edit routing-instances routing-instance-name protocols vpls associate-profile]
Release Information	Statement introduced in Junos OS Release 11.1.
Description	Specify the variable set to apply to the dynamic profile for the routing instance.
Options	<i>variable-set-name</i> —Name of the variable set to use when this dynamic profile is applied to the routing instance. You define this variable set at the [edit dynamic-profiles <i>profile-name</i>] hierarchy level.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Attaching the Dynamic Profiles to Routing Instances for the Complex Configuration on page 13

PART 3

Verification

- [Verifying Your Layer 2 Traffic Policers on VPLS Connections on page 21](#)

CHAPTER 5

Verifying Your Layer 2 Traffic Policers on VPLS Connections

- Verifying Your Layer 2 Traffic Policers on VPLS Connections on page 21

Verifying Your Layer 2 Traffic Policers on VPLS Connections

Purpose Display VPLS connections to verify that the dynamic profile is running on the Layer 2 VPN connection.

Action user@host> show vpls connections
Layer-2 VPN connections:

```
...
Instance: vpls-10gige
Local site: 10Gige-PE (2)
  connection-site      Type  St      Time last up      # Up trans
  1                    rmt   Up       Mar 28 21:27:57 2010      1
    Remote PE: 10.10.1.1, Negotiated control-word: No
    Incoming label: 262145, Outgoing label: 262146
    Local interface: lsi.1048576, Status: Up, Encapsulation: VPLS
    Dynamic profile: pw-policer
    Description: Intf - vpls vpls-10gige local site 2 remote site 1
```

Meaning The Dynamic profile field displays the policer that is currently running on the VPLS connection.

