

Network Configuration Example

Configuring Active Flow Monitoring Version 9

Release
11.2



Published: 2011-06-21

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.

Network Configuration Example Configuring Active Flow Monitoring Version 9

Release 11.2

Copyright © 2011, Juniper Networks, Inc.

All rights reserved.

Revision History

April 2011—R1 Junos OS 11.2

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

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. The 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

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

Flow Monitoring Overview	1
Active Flow Monitoring Overview	3
Active Flow Monitoring Applications	5
Best Practices for Configuring Active Flow Monitoring Version 9	7
Active and Inactive Timeouts	7
Sampling Rate	8
Sampling Run Length	8
Example: Configuring Active Flow Monitoring Version 9 for IPv4	9
Example: Configuring Active Flow Monitoring Version 9 for IPv6	17
Example: Configuring Active Flow Monitoring Version 9 for MPLS	25
Example: Configuring Active Flow Monitoring Version 9 for MPLS and IPv4	33
Example: Configuring Active Flow Monitoring Version 9 for IPv4, MPLS, and IPv6	43
Verifying Active Flow Monitoring Version 9	57
Verifying That Active Flow Monitoring Is Working	57
Verifying That the Services PIC Is Operational for Active Flow Monitoring . . .	57
Verifying That Sampling Is Enabled and the Filter Direction Is Correct for Active Flow Monitoring	58
Verifying That the Sampling Instance Is Applied to the Correct FPC for Active Flow Monitoring	59
Verifying That the Route Record Is Being Created for Active Flow Monitoring	59
Verifying That the Sampling Process Is Running for Active Flow Monitoring	60
Verifying That the TCP Connection Is Operational for Active Flow Monitoring	60
Verifying That the Services PIC Memory Is Not Overloaded for Active Flow Monitoring	61
Verifying That the Active Flow Monitoring Flow Collector Is Reachable	61

Flow Monitoring Overview

The flow monitoring application performs traffic flow monitoring and enables lawful interception of packets transiting between two routers. Traffic flows can either be passively monitored by an offline router or actively monitored by a router participating in the network.

Using a Juniper Networks router, a selection of PICs for M Series and T Series routers—including the Monitoring Services PIC, Monitoring Services II PIC, Adaptive Services PIC, and MultiServices PICs—and other networking hardware, you can monitor traffic flow and export the monitored traffic. Monitoring traffic allows you to do the following:

- Gather and export detailed information about traffic flows between source and destination routers in your network.
- Sample all incoming traffic on the monitoring interface and present the data in record format.
- Encrypt or tunnel outgoing records, intercepted traffic, or both.
- Direct filtered traffic to different packet analyzers and present the data in its original format.
- Intercept unwanted traffic, discard it, and perform accounting on the discarded packets.

There are two main types of flow monitoring:

- Active Flow Monitoring
- Passive Flow Monitoring

Related Documentation

- Active Flow Monitoring Overview on page 3
- Passive Flow Monitoring Overview

Active Flow Monitoring Overview

Flow monitoring versions 5, 8, and 9 support active flow monitoring. For active flow monitoring, the monitoring station participates in the network as an active router. The major actions the router can perform during active flow monitoring are as follows:

- Sampling—The router selects and analyzes only a portion of the traffic.
- Sampling with templates—The router selects, analyzes, and arranges a portion of the traffic into templates.
- Sampling per sampling instance—The router selects, analyzes, and arranges a portion of the traffic according to the configuration and binding of a sampling instance.
- Port mirroring—The router copies entire packets and sends the copies to another interface.
- Multiple port mirroring—The router sends multiple copies of monitored packets to multiple export interfaces with the **next-hop-group** statement at the **[edit forwarding-options]** hierarchy level.
- Discard accounting—The router accounts for selected traffic before discarding it. Such traffic is not forwarded out of the router. Instead, the traffic is quarantined and deleted.
- Flow-tap processing—The router processes requests for active flow monitoring dynamically by using the Dynamic Tasking Control Protocol (DTCP).

Related Documentation

- Flow Monitoring Overview on page 1
- Passive Flow Monitoring Overview

Active Flow Monitoring Applications

Flow monitoring can be used for many different reasons such as network planning, accounting, usage-based network billing, security, and monitoring for Denial-of-Service attacks.

Some examples of the types of things you can use flow monitoring for are:

- Tracking what kind of traffic is entering or exiting an ISP or corporate network.
- Tracking traffic flows between BGP autonomous systems.
- Tracking traffic flows between enterprise network regions.
- Taking a snapshot of the existing quality-of-service (QoS) policy results prior to making changes in QoS policy in case you need to roll back changes later in the process.
- Verifying that load balancing techniques are performing as intended.
- Capturing a base line of current network performance prior to making changes intended to improve performance so that you know if the changes are helping.
- Discovering if network users at an enterprise are using bandwidth for work-related activities or for non work-related activities.

Examples of how flow monitoring helps with network administration include the following:

- A large service provider uses active flow monitoring on its core uplinks as a way to collect data on the protocols in use, packet sizes, and flow durations to better understand the usage of its Internet service offering. This helps the provider understand where network growth is coming from.
- Service providers bill customers for the data sent or bandwidth used by sending captured flow data to third-party billing software.
- At a large enterprise, VoIP users at a remote site complained of poor voice quality. The flow monitoring reports showed that the VoIP traffic did not have the correct type of service settings.
- Users on an enterprise network, reported network slowdowns. The flow monitoring reports showed that one user's PC was generating a large portion of the network traffic. The PC was infected with malware.
- A growing enterprise planned to deploy new business management software and needed to know what type of network bandwidth demand the new software would create. During the software trial period, flow monitoring reports were used to identify the expected increase in traffic.

Thus, while flow monitoring is traditionally associated with traffic analysis, it also has a role in accounting and security.

Related Documentation

- Flow Monitoring Overview on page 1
- Active Flow Monitoring Overview on page 3

Best Practices for Configuring Active Flow Monitoring Version 9

Four settings control the behavior of active flow monitoring: Sampling rate, sampling run-length, flow active timeout, and flow inactive timeout. When you tune these settings, consider the following:

- Choosing a higher sampling rate or higher run-length increases the load on the services PIC.
- Selecting a higher sampling rate collects more information and provides finer grain flow information.
- A nonzero run-length provides trailing context regarding the packets immediately following a triggered sample.
- Selecting a larger active or inactive timeout value reduces the load on the export CPU and reduces the rate of packets going to the flow collector.

Active and Inactive Timeouts

A flow is considered inactive if a packet matching the filter is not received for a duration longer than the inactive timeout value.

Flow monitoring tracks flows as unidirectional streams of packets. It is not aware of application-level session properties or protocol details. However, there is some minimal awareness of TCP/IP properties. A flow is considered inactive when a TCP FIN, FIN-ACK, or RST control signal is received.

When the inactive timeout is triggered, the services PIC purges the flow from its flow table and generates an export record for the flow.

The inactive timeout can be set to as small a value as can be handled considering the load on the services PIC. The inactive timeout is typically several seconds (30 to 60 seconds). The administrator can tune the timeout to a larger value to try to reduce the load on the control CPU. The effectiveness of this setting for reducing CPU load depends on the overall input flow rate and the rate at which flows are expiring.

In a similar manner, an active timeout is triggered when the active timer expires and the flow is still active. The active timeout is intended to capture information about long-lived flows.

In the absence of an active timeout mechanism, a collector might not receive any information about a flow until it expires due to inactivity. Hence, the goal is to send periodic updates about a flow that has not expired.

When an active timeout is triggered, the flow start timestamp is not reset. Therefore, the collector can correlate a sequence of active timeout export packets and use the start time to identify long-lived flows, such as bulk transfers like FTP and peer-to-peer downloads of large files.

It is recommended having a value higher than the default for active timeout. Typical settings are in the range of several minutes (up to 10 minutes).

Sampling Rate

There is extensive research that helps identify the best choice for a sampling rate. Duffield et al ("Properties and Prediction of Flow Statistics from Sampled Packet Streams", ACM SIGCOMM 2002) consider a variety of objectives and recommend heuristics and formulas to compute the sampling rate.

If the objective is to obtain an accurate measurement of the original number of packets, the error in an estimate derived from sampled packets reduces in proportion to the square root of the sampling rate. For example, if the sampling rate is 100 and the original number of packets is 1 million, the expected error is on the order of $(100/1,000,000)$ or about 1 percent. In other words, if the sampled packet count is 10,000, the original packet count can be in the range 990,000 to 1.01 million. This agrees with the idea that a higher sampling rate reduces the error in estimation.

Sampling Run Length

The run-length statement specifies the number of matching packets to sample following the initial one-packet trigger event. By default, the run length is 0, which means that no more traffic is sampled after the trigger event. The range is from 0 through 20 packets. Configuring a run length greater than 0 allows you to sample packets following those already being sampled.

Related Documentation

- Flow Monitoring Overview on page 1
- Active Flow Monitoring Overview on page 3
- Active Flow Monitoring Applications on page 5
- Example: Configuring Active Flow Monitoring Version 9 for IPv4 on page 9
- Example: Configuring Active Flow Monitoring Version 9 for IPv6 on page 17
- Example: Configuring Active Flow Monitoring Version 9 for MPLS on page 25
- Example: Configuring Active Flow Monitoring Version 9 for MPLS and IPv4 on page 33
- Example: Configuring Active Flow Monitoring Version 9 for IPv4, MPLS, and IPv6 on page 43
- Verifying Active Flow Monitoring Version 9 on page 57

Example: Configuring Active Flow Monitoring Version 9 for IPv4

This example shows how to monitor IPv4 flows by using active flow Monitoring version 9. It is organized in the following sections:

- Requirements on page 9
- Overview of Flow Monitoring on page 9
- Configuring Active Flow Monitoring Version 9 for IPv4 on page 9

Requirements

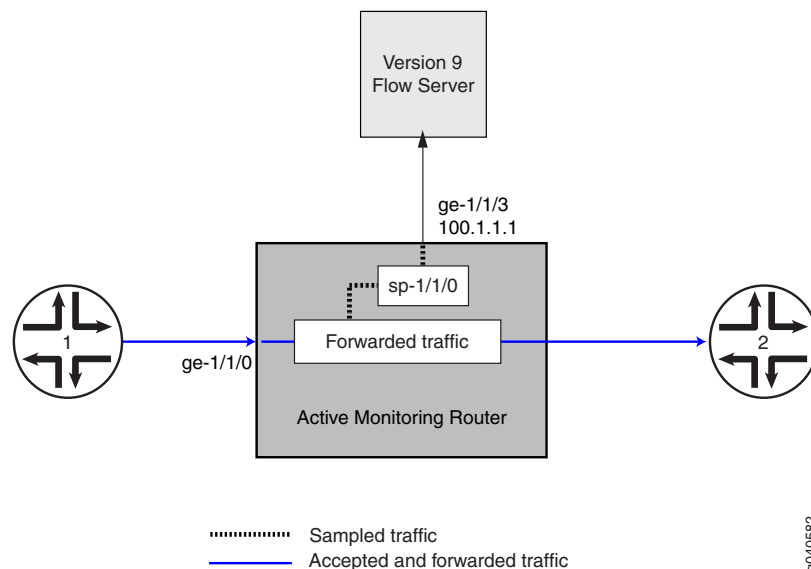
This example requires the following hardware and software components:

- Junos OS Release 9.2 or later
- One M40e or M320 Multiservice Edge Router, MX Series 3D Universal Edge Routers, or T Series Core Router
- One Adaptive Services PIC

Overview of Flow Monitoring

This example explains how to monitor IPv4 flows.

The physical connections used in this example are shown in Figure 1 on page 9.



Configuring Active Flow Monitoring Version 9 for IPv4

- | | |
|-------------------------------|---|
| Step-by-Step Procedure | <ol style="list-style-type: none">1. Enable the services PIC interface to process IPv4 addresses by including the family statement and specifying the inet option at the [edit interfaces sp-1/1/0 unit 0] hierarchy level.

<pre>[edit interfaces] sp-1/1/0 {</pre> |
|-------------------------------|---|

```
    unit 0 {  
        family inet;  
    }  
}
```

2. Configure the interface connected to the flow collector by including the **address** statement and specifying **100.1.1.1/24** as the IPv4 address of the interface at the **[edit interfaces ge-1/1/3 unit 0 family inet]** hierarchy level.

```
[edit interfaces]  
ge-1/1/3 {  
    description to-flow-collector;  
    unit 0 {  
        family inet {  
            address 100.1.1.1/24;  
        }  
    }  
}
```

3. Create a version 9 template by including the **template** statement and specifying **v4_template** as the name of the template at the **[edit services flow-monitoring version9]** hierarchy level.

Enable the template for IPv4 flows by including the **ipv4-template** statement at the **[edit services flow-monitoring version9 template v4_template]** hierarchy level.

Configure the flow active timeout by including the **flow-active-timeout** statement and specifying **600** seconds at the **[edit services flow-monitoring version9 template v4_template]** hierarchy level. Configure the flow inactive timeout by including the **flow-inactive-timeout** statement and specifying **30** seconds at the **[edit services flow-monitoring version9 template v4_template]** hierarchy level.

```
[edit services]  
flow-monitoring {  
    version9 {  
        template v4_template {  
            flow-active-timeout 600;  
            flow-inactive-timeout 30;  
            ipv4-template;  
        }  
    }  
}
```

4. Configure the rate at which the router sends template definitions and options to the flow collector for IPv4.

Since version 9 flow monitoring traffic is unidirectional from the monitor (router) to the flow collector, configure the monitor to send template definitions and options, such as sampling rate, to the collector.

In this example, the template definitions and options are refreshed every 600 seconds or 480000 packets, whichever occurs first.

Include the **packets** statement and specify **480000** packets at the **[edit services flow-monitoring version9 template v4_template template-refresh-rate]** and **[edit services flow-monitoring version9 template v4_template option-refresh-rate]** hierarchy levels. Include the **seconds** statement and specify **600** seconds at the **[edit services**

flow-monitoring template v4_template version9 template-refresh-rate] and **[edit services flow-monitoring version9 template v4_template option-refresh-rate]** hierarchy levels.

```
[edit services flow-monitoring version9 template v4_template]
template-refresh-rate {
  packets 480000;
  seconds 600;
}
option-refresh-rate {
  packets 480000;
  seconds 600;
}
```

5. Configure the sampling rate and run length.

The sampling rate determines the ratio of the number of packets to be sampled. For example, if you specify a rate of 10, 1 out of every 10 packets is sampled. In this example, the rate is 1 out of every 1 packets.

Sampling can be configured as a global chassis configuration that is applicable to all Flexible PIC Concentrators (FPCs) and Dense Port Concentrators (DPCs) at the **[edit forwarding-options sampling input]** hierarchy level. Sampling can also be configured at the **[edit forwarding-options sampling instance *instance-name*]** hierarchy level and then applied to a single FPC.

The run length sets the number of samples to be taken following the initial trigger event. This allows you to sample packets following those already being sampled. Since you are sampling every packet in this example, the run length can be set to 1.

To configure the sampling rate, include the **rate** statement and specify 1 as the rate at the **[edit forwarding-options sampling instance ins1 input]** hierarchy level. To configure the run length, include the **run-length** statement and specify 1 as the run length at the **[edit forwarding-options sampling instance ins1 input]** hierarchy level.

```
[edit forwarding-options]
sampling {
  instance ins1 {
    input {
      rate 1;
      run-length 1;
    }
  }
}
```

6. Apply the sampling instance to the desired FPC or DPC.

The FPC number must match the FPC portion of the interface name for the interface on which sampling is enabled.

To apply the sampling instance, include the **sampling-instance** statement and specify **ins1** at the **[edit chassis fpc 1]** hierarchy level.

```
[edit]
chassis {
  fpc 1 {
    sampling-instance ins1;
  }
}
```

```
}  
}
```

7. Configure the flow collector and enable active flow monitoring using the version 9 template format.

To configure the flow collector, include the **flow-server** statement and specify **100.1.1.2** as the IPv4 address of the host system that is collecting traffic flows using version 9 at the **[edit forwarding-options sampling instance ins1 family inet output]** hierarchy level. Also include the **port** statement and specify UDP port **2055** for use by the flow collector.

To enable active flow monitoring using the version 9 template format, include the **template** statement and specify **v4_template** as the name of the template to use at the **[edit forwarding-options sampling instance ins1 family inet output flow-server 100.1.1.2 version9]** hierarchy level.

```
[edit forwarding-options sampling instance ins1]  
family inet {  
  output {  
    flow-server 100.1.1.2 {  
      port 2055;  
      version9 {  
        template v4_template;  
      }  
    }  
  }  
}
```

8. Configure the IPv4 source address for the services PIC to be used in flow export.

To configure the IPv4 source address for the **sp-1/1/0** interface, include the **source-address** statement and specify **12.1.1.1** at the **[edit forwarding-options sampling instance ins1 family inet output interface sp-1/1/0]** hierarchy level.

```
[edit forwarding-options sampling instance ins1 family inet output]  
interface sp-1/1/0 {  
  source-address 12.1.1.1;  
}
```

9. Configure the firewall filter.

The firewall filter identifies the traffic flows that need to be sampled and processed by the services PIC. Note that the implied “from” clause in the filter determines the packets that are matched and sampled according to the sampling rate.

To configure the firewall filter, include the **filter** statement and specify **ipv4_sample_filter** as the name of the filter at the **[edit firewall family inet]** hierarchy level. Include the **term** statement and specify **1** as the name of the term. For active monitoring using version 9, you must include the **sample** and **accept** action statements at the **[edit firewall family inet filter ipv4_sample_filter term 1 then]** hierarchy level.

```
[edit firewall]  
family inet {  
  filter ipv4_sample_filter {  
    term 1 {
```

```

        then {
            sample;
            accept;
        }
    }
}

```

10. Apply the firewall filter to the set of media interfaces where traffic flow needs to be sampled.

The filter can be applied to either the ingress or egress traffic depending on the use case. In this example, the filter is applied to the ingress (input) traffic.

To apply the firewall filter to the **ge-1/1/0** interface, include the **input** statement and specify **ipv4_sample_filter** as the name of the filter at the **[edit interfaces ge-1/1/0 unit 0 family inet filter]** hierarchy level.

```

[edit]
interfaces {
  ge-1/1/0 {
    unit 0 {
      family inet {
        filter {
          input ipv4_sample_filter;
        }
      }
    }
  }
}

```

Results For your reference, the relevant sample configuration for the IPv4 flow collector follows.

```

[edit]
services {
  flow-monitoring {
    version9 {
      template v4_template {
        flow-active-timeout 600;
        flow-inactive-timeout 30;
        template-refresh-rate {
          packets 480000;
          seconds 600;
        }
        option-refresh-rate {
          packets 480000;
          seconds 600;
        }
      }
      ipv4-template;
    }
  }
}
forwarding-options {
  sampling {
    instance ins1 {

```

```
    input {
      rate 1;
      run-length 1;
    }
    family inet {
      output {
        flow-server 100.1.1.2 {
          port 2055;
          version9 {
            template v4_template;
          }
        }
        interface sp-1/1/0 {
          source-address 12.1.1.1;
        }
      }
    }
  }
}
chassis {
  fpc 1 {
    sampling-instance ins1;
  }
}
firewall {
  family inet {
    filter ipv4_sample_filter {
      term 1 {
        then {
          sample;
          accept;
        }
      }
    }
  }
}
interfaces {
  ge-1/1/0 {
    description media-interface-for-sampling;
    unit 0 {
      family inet {
        filter {
          input ipv4_sample_filter;
        }
      }
    }
  }
  sp-1/1/0 {
    description sampling-services-pic;
    unit 0 {
      family inet;
    }
  }
  ge-1/1/3 {
    description to-flow-collector;
  }
}
```

```
unit 0 {  
    family inet {  
        address 100.1.1.1/24;  
    }  
}  
}
```

**Related
Documentation**

- [Flow Monitoring Overview on page 1](#)
- [Active Flow Monitoring Overview on page 3](#)
- [Active Flow Monitoring Applications on page 5](#)
- [Best Practices for Configuring Active Flow Monitoring Version 9 on page 7](#)
- [Example: Configuring Active Flow Monitoring Version 9 for IPv6 on page 17](#)
- [Example: Configuring Active Flow Monitoring Version 9 for MPLS on page 25](#)
- [Example: Configuring Active Flow Monitoring Version 9 for MPLS and IPv4 on page 33](#)
- [Example: Configuring Active Flow Monitoring Version 9 for IPv4, MPLS, and IPv6 on page 43](#)
- [Verifying Active Flow Monitoring Version 9 on page 57](#)

Example: Configuring Active Flow Monitoring Version 9 for IPv6

This example shows how to monitor IPv6 flows by using active flow Monitoring version 9. It is organized in the following sections:

- Requirements on page 17
- Overview of Flow Monitoring on page 17
- Configuring Active Flow Monitoring Version 9 for IPv6 on page 17

Requirements

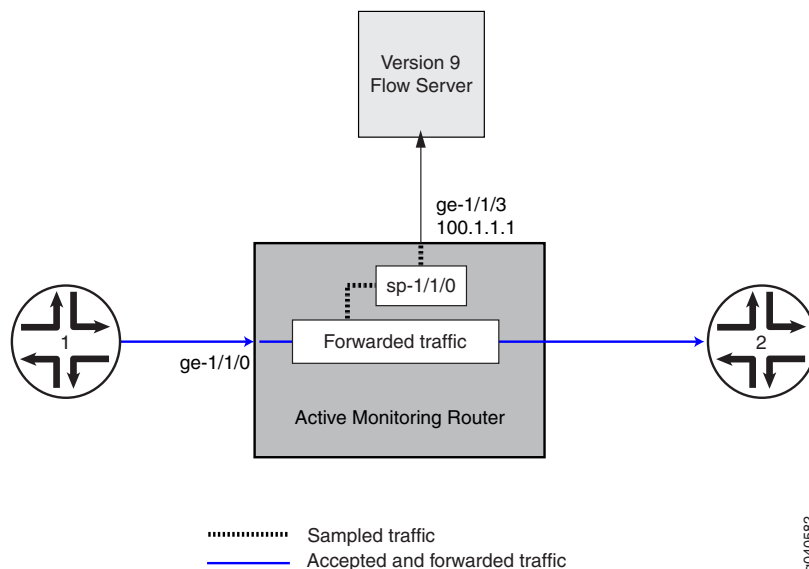
This example requires the following hardware and software components:

- Junos OS Release 9.2 or later
- One M Series Multiservice Edge Router, MX Series 3D Universal Edge Routers, or T Series Core Router
- One Adaptive Services PIC

Overview of Flow Monitoring

This example explains how to monitor IPv6 flows.

The physical connections used in this example are shown in Figure 2 on page 17.



Configuring Active Flow Monitoring Version 9 for IPv6

- | | |
|-------------------------------|--|
| Step-by-Step Procedure | <ol style="list-style-type: none">1. Enable the services PIC interface to process IPv6 addresses by including the family statement and specifying the inet6 option at the [edit interfaces sp-1/1/0 unit 0] hierarchy level.

<pre>[edit interfaces] sp-1/1/0 {</pre> |
|-------------------------------|--|

```
    unit 0 {  
        family inet6;  
    }  
}
```

2. Configure the interface connected to the flow collector by including the **address** statement and specifying **100.1.1.1/24** as the IPv4 address of the interface at the **[edit interfaces ge-1/1/3 unit 0 family inet]** hierarchy level.

```
[edit interfaces]  
ge-1/1/3 {  
    description to-flow-collector;  
    unit 0 {  
        family inet {  
            address 100.1.1.1/24;  
        }  
    }  
}
```

3. Create a version 9 template by including the **template** statement and specifying **v6_template** as the name of the template at the **[edit services flow-monitoring version9]** hierarchy level.

Enable the template for IPv6 flows by including the **ipv6-template** statement at the **[edit services flow-monitoring version9 template v6_template]** hierarchy level.

Configure the flow active timeout by including the **flow-active-timeout** statement and specifying **600** seconds at the **[edit services flow-monitoring version9 template v6_template]** hierarchy level. Configure the flow inactive timeout by including the **flow-inactive-timeout** statement and specifying **30** seconds at the **[edit services flow-monitoring version9 template v6_template]** hierarchy level.

```
[edit services]  
flow-monitoring {  
    version9 {  
        template v6_template {  
            flow-active-timeout 600;  
            flow-inactive-timeout 30;  
            ipv6-template;  
        }  
    }  
}
```

4. Configure the rate at which the router sends template definitions and options to the flow collector for IPv6.

Since version 9 flow monitoring traffic is unidirectional from the monitor (router) to the flow collector, configure the monitor to send template definitions and options, such as sampling rate, to the collector.

In this example, the template definitions and options are refreshed every 600 seconds or 480000 packets, whichever occurs first.

Include the **packets** statement and specify **480000** packets at the **[edit services flow-monitoring version9 template v6_template template-refresh-rate]** and **[edit services flow-monitoring version9 template v6_template option-refresh-rate]** hierarchy levels. Include the **seconds** statement and specify **600** seconds at the **[edit services**

flow-monitoring version9 template v6_template template-refresh-rate] and **[edit services flow-monitoring version9 template v6_template option-refresh-rate]** hierarchy levels.

```
[edit services flow-monitoring version9 template v6_template]
template-refresh-rate {
  packets 480000;
  seconds 600;
}
option-refresh-rate {
  packets 480000;
  seconds 600;
}
```

5. Configure the sampling rate and run length.

The sampling rate determines the ratio of the number of packets to be sampled. For example, if you specify a rate of 10, 1 out of every 10 packets is sampled. In this example, the rate is 1 out of every 1 packets.

Sampling can be configured as a global chassis configuration that is applicable to all Flexible PIC Concentrators (FPCs) and Dense Port Concentrators (DPCs) at the **[edit forwarding-options sampling input]** hierarchy level. Sampling can also be configured at the **[edit forwarding-options sampling instance *instance-name*]** hierarchy level and then applied to a single FPC.

The run length sets the number of samples to be taken following the initial trigger event. This allows you to sample packets following those already being sampled. Since you are sampling every packet in this example, the run length can be set to 1.

To configure the sampling rate, include the **rate** statement and specify 1 as the rate at the **[edit forwarding-options sampling instance ins1 input]** hierarchy level. To configure the run length, include the **run-length** statement and specify 1 as the run length at the **[edit forwarding-options sampling instance ins1 input]** hierarchy level.

```
[edit forwarding-options]
sampling {
  instance ins1 {
    input {
      rate 1;
      run-length 1;
    }
  }
}
```

6. Apply the sampling instance to the desired FPC or DPC.

The FPC number must match the FPC portion of the interface name for the interface on which sampling is enabled.

To apply the sampling instance, include the **sampling-instance** statement and specify **ins1** at the **[edit chassis fpc 1]** hierarchy level.

```
[edit]
chassis {
  fpc 1 {
    sampling-instance ins1;
```

```
}  
}
```

7. Configure the flow collector and enable active flow monitoring using the version 9 template format.

To configure the flow collector, include the **flow-server** statement and specify **100.1.1.2** as the IPv4 address of the host system that is collecting traffic flows using version 9 at the **[edit forwarding-options sampling instance ins1 family inet6 output]** hierarchy level. Also include the **port** statement and specify UDP port **2055** for use by the flow collector.

To enable active flow monitoring using the version 9 template format, include the **template** statement and specify **v6_template** as the name of the template to use at the **[edit forwarding-options sampling instance ins1 family inet6 output flow-server 100.1.1.2 version9]** hierarchy level.

```
[edit forwarding-options sampling instance ins1]  
family inet6 {  
  output {  
    flow-server 100.1.1.2 {  
      port 2055;  
      version9 {  
        template v6_template;  
      }  
    }  
  }  
}
```

8. Configure the IPv4 source address for the services PIC to be used in flow export.

To configure the IPv4 source address for the **sp-1/1/0** interface, include the **source-address** statement and specify **12.1.1.1** at the **[edit forwarding-options sampling instance ins1 family inet6 output interface sp-1/1/0]** hierarchy level.

```
[edit forwarding-options sampling instance ins1 family inet6 output]  
interface sp-1/1/0 {  
  source-address 12.1.1.1;  
}
```

9. Configure the firewall filter.

The firewall filter identifies the traffic flows that need to be sampled and processed by the services PIC. Note that the implied “from” clause in the filter determines the packets that are matched and sampled according to the sampling rate.

To configure the firewall filter, include the **filter** statement and specify **ipv6_sample_filter** as the name of the filter at the **[edit firewall family inet6]** hierarchy level. Include the **term** statement and specify **1** as the name of the term. For active monitoring using version 9, you must include the **sample** and **accept** action statements at the **[edit firewall family inet6 filter ipv6_sample_filter term 1 then]** hierarchy level.

```
[edit firewall]  
family inet6 {  
  filter ipv6_sample_filter {  
    term 1 {
```

```

        then {
            sample;
            accept;
        }
    }
}

```

10. Apply the firewall filter to the set of media interfaces where traffic flow needs to be sampled.

The filter can be applied to either the ingress or egress traffic depending on the use case. In this example, the filter is applied to the egress (output) traffic.

To apply the firewall filter to the **ge-1/1/0** interface, include the **output** statement and specify **ipv6_sample_filter** as the name of the filter at the **[edit interfaces ge-1/1/0 unit 0 family inet filter]** hierarchy level.

```

[edit]
interfaces {
  ge-1/1/0 {
    unit 0 {
      family inet6 {
        filter {
          output ipv6_sample_filter;
        }
      }
    }
  }
}

```

Results For your reference, the relevant sample configuration for the IPv6 flow collector follows.

```

[edit]
services {
  flow-monitoring {
    version9 {
      template v6_template {
        flow-active-timeout 600;
        flow-inactive-timeout 30;
        template-refresh-rate {
          packets 480000;
          seconds 600;
        }
        option-refresh-rate {
          packets 480000;
          seconds 600;
        }
      }
      ipv6-template;
    }
  }
}
forwarding-options {
  sampling {
    instance ins1 {

```

```
    input {
      rate 1;
      run-length 1;
    }
    family inet6 {
      output {
        flow-server 100.1.1.2 {
          port 2055;
          version9 {
            template v6_template;
          }
        }
        interface sp-1/1/0 {
          source-address 12.1.1.1;
        }
      }
    }
  }
}
chassis {
  fpc 1 {
    sampling-instance ins1;
  }
}
firewall {
  family inet6 {
    filter ipv6_sample_filter {
      term 1 {
        then {
          sample;
          accept;
        }
      }
    }
  }
}
interfaces {
  ge-1/1/0 {
    description media-interface-for-sampling;
    unit 0 {
      family inet6 {
        filter {
          output ipv6_sample_filter;
        }
      }
    }
  }
  sp-1/1/0 {
    description sampling-services-pic;
    unit 0 {
      family inet6;
    }
  }
  ge-1/1/3 {
    description to-flow-collector;
  }
}
```

```
unit 0 {  
    family inet {  
        address 100.1.1.1/24;  
    }  
}  
}
```

**Related
Documentation**

- [Flow Monitoring Overview on page 1](#)
- [Active Flow Monitoring Overview on page 3](#)
- [Active Flow Monitoring Applications on page 5](#)
- [Best Practices for Configuring Active Flow Monitoring Version 9 on page 7](#)
- [Example: Configuring Active Flow Monitoring Version 9 for IPv4 on page 9](#)
- [Example: Configuring Active Flow Monitoring Version 9 for MPLS on page 25](#)
- [Example: Configuring Active Flow Monitoring Version 9 for MPLS and IPv4 on page 33](#)
- [Example: Configuring Active Flow Monitoring Version 9 for IPv4, MPLS, and IPv6 on page 43](#)
- [Verifying Active Flow Monitoring Version 9 on page 57](#)

Example: Configuring Active Flow Monitoring Version 9 for MPLS

This example shows how to monitor MPLS flows by using active flow Monitoring version 9. It is organized in the following sections:

- Requirements on page 25
- Overview of Flow Monitoring on page 25
- Configuring Active Flow Monitoring Version 9 for MPLS on page 25

Requirements

This example requires the following hardware and software components:

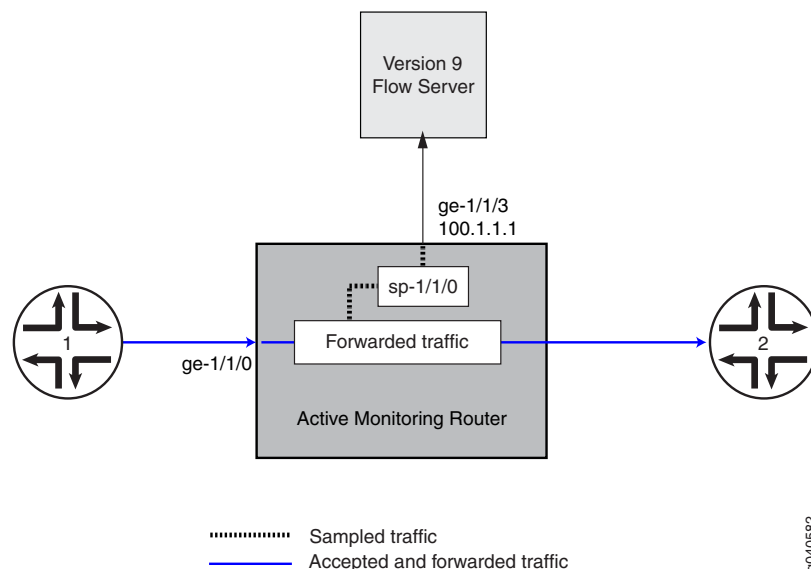
- Junos OS Release 9.2 or later
- One M Series Multiservice Edge Router, MX Series 3D Universal Edge Routers, or T Series Core Router
- One Adaptive Services PIC

Overview of Flow Monitoring

This example explains how to monitor MPLS flows.

The physical connections used in this example are shown in Figure 3 on page 25.

Figure 3: Active Flow Monitoring Version 9 for MPLS Topology



Configuring Active Flow Monitoring Version 9 for MPLS

- Step-by-Step Procedure**
1. Enable the services PIC interface to process MPLS addresses by including the **family** statement and specifying the **mpls** option at the **[edit interfaces sp-1/1/0 unit 0]** hierarchy level.
[edit interfaces]

```
sp-1/1/0 {  
  unit 0 {  
    family mpls;  
  }  
}
```

2. Configure the interface connected to the flow collector by including the **address** statement and specifying **100.1.1.1/24** as the IPv4 address of the interface at the **[edit interfaces ge-1/1/3 unit 0 family inet]** hierarchy level.

```
[edit interfaces]  
ge-1/1/3 {  
  description to-flow-collector;  
  unit 0 {  
    family inet {  
      address 100.1.1.1/24;  
    }  
  }  
}
```

3. Create a version 9 template by including the **template** statement and specifying **mpls** as the name of the template at the **[edit services flow-monitoring version9]** hierarchy level.

Enable the template for MPLS flows by including the **mpls-template** statement at the **[edit services flow-monitoring version9 template mpls]** hierarchy level. Also include the **label-position** statement and specify label positions 1 and 2 at the **[edit services flow-monitoring version9 template mpls mpls-template]** hierarchy level.

Configure the flow active timeout by including the **flow-active-timeout** statement and specifying **600** seconds at the **[edit services flow-monitoring version9 template mpls]** hierarchy level. Configure the flow inactive timeout by including the **flow-inactive-timeout** statement and specifying **30** seconds at the **[edit services flow-monitoring version9 template mpls]** hierarchy level.

```
[edit services]  
flow-monitoring {  
  version9 {  
    template mpls {  
      flow-active-timeout 600;  
      flow-inactive-timeout 30;  
      mpls-template {  
        label-position [ 1 2 ];  
      }  
    }  
  }  
}
```

4. Configure the rate at which the router sends template definitions and options to the flow collector for MPLS.

Since version 9 flow monitoring traffic is unidirectional from the monitor (router) to the flow collector, configure the monitor to send template definitions and options, such as sampling rate, to the collector.

In this example, the template definitions and options are refreshed every 600 seconds or 480000 packets, whichever occurs first.

Include the **packets** statement and specify **480000** packets at the **[edit services flow-monitoring version9 template mpls template-refresh-rate]** and **[edit services flow-monitoring version9 template mpls option-refresh-rate]** hierarchy levels. Include the **seconds** statement and specify **600** seconds at the **[edit services flow-monitoring version9 template mpls template-refresh-rate]** and **[edit services flow-monitoring version9 template mpls option-refresh-rate]** hierarchy levels.

```
[edit services flow-monitoring version9 template mpls]
template-refresh-rate {
  packets 480000;
  seconds 600;
}
option-refresh-rate {
  packets 480000;
  seconds 600;
}
```

5. Configure the sampling rate and run length.

The sampling rate determines the ratio of the number of packets to be sampled. For example, if you specify a rate of 10, 1 out of every 10 packets is sampled. In this example, the rate is 1 out of every 1 packets.

Sampling can be configured as a global chassis configuration that is applicable to all Flexible PIC Concentrators (FPCs) and Dense Port Concentrators (DPCs) at the **[edit forwarding-options sampling input]** hierarchy level. Sampling can also be configured at the **[edit forwarding-options sampling instance *instance-name*]** hierarchy level and then applied to a single FPC.

In this example two sampling categories are created. The global instance is configured to sample all packets matching a flow. Instance **inst1** is configured to sample one in every 10 packets.

To configure the global rate, include the **rate** statement and specify **1** as the rate at the **[edit forwarding-options sampling input]** hierarchy level. To configure the instance rate, include the **rate** statement and specify **10** as the rate at the **[edit forwarding-options sampling instance *inst1* input]** hierarchy level.

```
[edit forwarding-options]
sampling {
  input {
    rate 1;
  }
  instance inst1 {
    input {
      rate 10;
    }
  }
}
```

6. Apply the sampling instance to the desired FPC or DPC.

The FPC number must match the FPC portion of the interface name for the interface on which sampling is enabled.

To apply the sampling instance, include the **sampling-instance** statement and specify **ins1** at the **[edit chassis fpc 1]** hierarchy level.

```
[edit]
chassis {
  fpc 1 {
    sampling-instance ins1;
  }
}
```

7. Configure the flow collector and enable active flow monitoring using the version 9 template format.

To configure the flow collector, include the **flow-server** statement and specify the IP address of the host system that is collecting traffic flows using version 9 at the **[edit forwarding-options sampling instance ins1 family mpls output]** hierarchy level. Also include the **port** statement and specify UDP port **2055** for use by the flow collector.

To enable active flow monitoring using the version 9 template format, include the **template** statement and specify **mpls** as the name of the template to use at the **[edit forwarding-options sampling instance ins1 family mpls output flow-server 100.1.1.2 version9]** hierarchy level.

```
[edit forwarding-options sampling instance ins1]
family mpls {
  output {
    flow-server 100.1.1.2 {
      port 2055;
      version9 {
        template mpls;
      }
    }
  }
}
```

8. Configure the IPv4 source address for the services PIC to be used in flow export.

To configure the IPv4 source address for the **sp-1/1/0** interface, include the **source-address** statement and specify **12.1.1.1** at the **[edit forwarding-options sampling instance ins1 family mpls output interface sp-1/1/0]** hierarchy level.

```
[edit forwarding-options sampling instance ins1 family mpls output]
interface sp-1/1/0 {
  source-address 12.1.1.1;
}
```

9. Configure the firewall filter.

The firewall filter identifies the traffic flows that need to be sampled and processed by the services PIC. Note that the implied “from” clause in the filter determines the packets that are matched and sampled according to the sampling rate.

To configure the firewall filter, include the **filter** statement and specify **mpls_sample_filter** as the name of the filter at the **[edit firewall family inet]** hierarchy level. Include the **term** statement and specify **1** as the name of the term. For active monitoring using version 9, you must include the **sample** and **accept** action statements at the **[edit firewall family mpls filter mpls_sample_filter term 1 then]** hierarchy level.

```
[edit firewall]
family mpls {
  filter mpls_sample_filter {
    term 1 {
      then {
        sample;
        accept;
      }
    }
  }
}
```

10. Apply the firewall filter to the set of media interfaces where traffic flow needs to be sampled.

To apply the firewall filter to the **ge-1/1/0** interface, include the **input** statement and specify **mpls_sample_filter** as the name of the filter at the **[edit interfaces ge-1/1/0 unit 0 family mpls filter]** hierarchy level.

```
[edit]
interfaces {
  ge-1/1/0 {
    unit 0 {
      family mpls {
        filter {
          input mpls_sample_filter;
        }
      }
    }
  }
}
```

Results For your reference, the relevant sample configuration for the MPLS flow collector follows.

```
[edit]
services {
  flow-monitoring {
    version9 {
      template mpls {
        flow-active-timeout 600;
        flow-inactive-timeout 30;
        template-refresh-rate {
          packets 480000;
          seconds 600;
        }
      }
      option-refresh-rate {
        packets 480000;
        seconds 600;
      }
    }
  }
}
```

```
    }
    mpls-template {
        label-position [ 1 2 ];
    }
}
}
}
}
forwarding-options {
    sampling {
        input {
            rate 1;
        }
        instance ins1 {
            input {
                rate 10;
            }
            family mpls {
                output {
                    flow-server 100.1.1.2 {
                        port 2055;
                        version9 {
                            template mpls {
                                }
                            }
                        }
                    interface sp-1/1/0 {
                        source-address 12.1.1.1;
                    }
                }
            }
        }
    }
}
}
}
}
chassis {
    fpc 1 {
        sampling-instance ins1;
    }
}
firewall {
    family mpls {
        filter mpls_sample_filter {
            term 1 {
                then {
                    sample;
                    accept;
                }
            }
        }
    }
}
}
}
interfaces {
    ge-1/1/0 {
        description media-interface-for-sampling;
        unit 0 {
            family mpls {
```

```
        filter {
            input mpls_sample_filter;
        }
    }
}
sp-1/1/0 {
    description sampling-services-pic;
    unit 0 {
        family mpls;
    }
}
ge-1/1/3 {
    description to-flow-collector;
    unit 0 {
        family inet {
            address 100.1.1.1/24;
        }
    }
}
```

**Related
Documentation**

- [Flow Monitoring Overview on page 1](#)
- [Active Flow Monitoring Overview on page 3](#)
- [Active Flow Monitoring Applications on page 5](#)
- [Best Practices for Configuring Active Flow Monitoring Version 9 on page 7](#)
- [Example: Configuring Active Flow Monitoring Version 9 for IPv4 on page 9](#)
- [Example: Configuring Active Flow Monitoring Version 9 for IPv6 on page 17](#)
- [Example: Configuring Active Flow Monitoring Version 9 for MPLS and IPv4 on page 33](#)
- [Example: Configuring Active Flow Monitoring Version 9 for IPv4, MPLS, and IPv6 on page 43](#)
- [Verifying Active Flow Monitoring Version 9 on page 57](#)

Example: Configuring Active Flow Monitoring Version 9 for MPLS and IPv4

This example shows how to monitor MPLS and IPv4 flows by using active flow Monitoring version 9. It is organized in the following sections:

- Requirements on page 33
- Overview of Flow Monitoring on page 33
- Configuring Active Flow Monitoring Version 9 for MPLS and IPv4 on page 33

Requirements

This example requires the following hardware and software components:

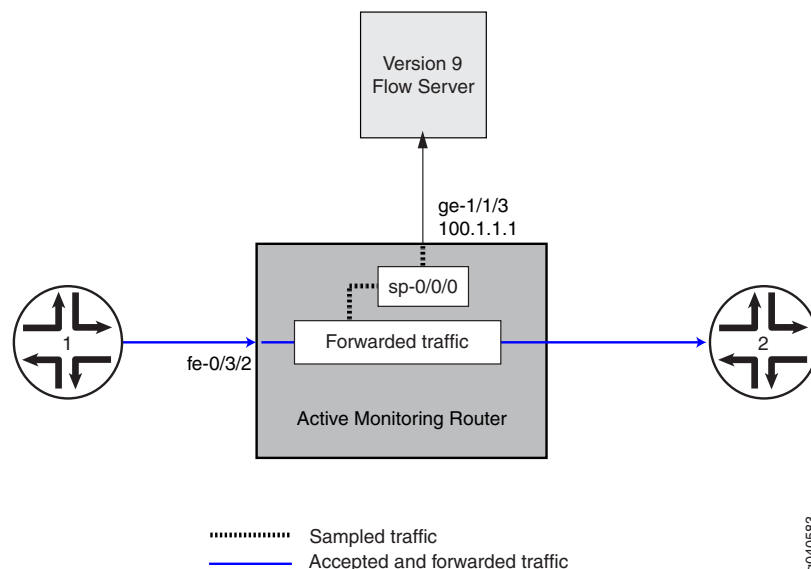
- Junos OS Release 9.2 or later
- One M Series Multiservice Edge Router, MX Series 3D Universal Edge Routers, or T Series Core Router
- One Adaptive Services PIC

Overview of Flow Monitoring

This example explains how to monitor MPLS and IPv4 flows.

The physical connections used in this example are shown in Figure 4 on page 33.

Figure 4: Active Flow Monitoring Version 9 for MPLS and IPv4 Topology



Configuring Active Flow Monitoring Version 9 for MPLS and IPv4

- | | |
|-------------------------------|---|
| Step-by-Step Procedure | <ol style="list-style-type: none">1. Enable the services PIC interface to process MPLS and IPv4 addresses by including the family statement and specifying the mpls option and the inet option at the [edit interfaces sp-0/0/0 unit 0] hierarchy level.

[edit interfaces] |
|-------------------------------|---|

```
sp-0/0/0 {  
  unit 0 {  
    family inet;  
    family mpls;  
  }  
}
```

2. Configure the interface connected to the flow collector by including the **address** statement and specifying **100.1.1.1/24** as the IPv4 address of the interface at the **[edit interfaces ge-1/1/3 unit 0 family inet]** hierarchy level.

```
[edit interfaces]  
ge-1/1/3 {  
  description to-flow-collector;  
  unit 0 {  
    family inet {  
      address 100.1.1.1/24;  
    }  
  }  
}
```

3. Create a version 9 template by including the **template** statement and specifying **mpls-ipv4** as the name of the template at the **[edit services flow-monitoring version9]** hierarchy level.

Enable the template for MPLS and IPv4 flows by including the **mpls-ipv4-template** statement at the **[edit services flow-monitoring version9 template mpls-ipv4]** hierarchy level. Also include the **label-position** statement and specify label positions **1** and **2** at the **[edit services flow-monitoring version9 template mpls-ipv4 mpls-ipv4-template]** hierarchy level.

Configure the flow active timeout by including the **flow-active-timeout** statement and specifying **600** seconds at the **[edit services flow-monitoring version9 template mpls-ipv4]** hierarchy level. Configure the flow inactive timeout by including the **flow-inactive-timeout** statement and specifying **30** seconds at the **[edit services flow-monitoring version9 template mpls-ipv4]** hierarchy level.

```
[edit services]  
flow-monitoring {  
  version9 {  
    template mpls-ipv4 {  
      flow-active-timeout 600;  
      flow-inactive-timeout 30;  
      mpls-ipv4-template {  
        label-position [ 1 2 ];  
      }  
    }  
  }  
}
```

4. Configure the rate at which the router sends template definitions and options to the flow collector for IPv4 and MPLS.

Since version 9 flow monitoring traffic is unidirectional from the monitor (router) to the flow collector, configure the monitor to send template definitions and options, such as sampling rate, to the collector.

In this example, the template definitions and options are refreshed every 600 seconds or 480000 packets, whichever occurs first.

Include the **packets** statement and specify **480000** packets at the **[edit services flow-monitoring version9 template mpls-ipv4 template-refresh-rate]** and **[edit services flow-monitoring version9 template mpls-ipv4 option-refresh-rate]** hierarchy levels. Include the **seconds** statement and specify **600** seconds at the **[edit services flow-monitoring version9 template mpls-ipv4 template-refresh-rate]** and **[edit services flow-monitoring version9 template mpls-ipv4 option-refresh-rate]** hierarchy levels.

```
[edit services flow-monitoring version9 template mpls-ipv4]
template-refresh-rate {
  packets 480000;
  seconds 600;
}
option-refresh-rate {
  packets 480000;
  seconds 600;
}
```

5. Configure the sampling rate and run length.

The sampling rate determines the ratio of the number of packets to be sampled. For example, if you specify a rate of 10, 1 out of every 10 packets is sampled. In this example, the rate is 1 out of every 1 packets.

Sampling can be configured as a global chassis configuration that is applicable to all Flexible PIC Concentrators (FPCs) and Dense Port Concentrators (DPCs) at the **[edit forwarding-options sampling input]** hierarchy level. Sampling can also be configured at the **[edit forwarding-options sampling instance *instance-name*]** hierarchy level and then applied to a single FPC.

In this example, two sampling categories are created. The global instance is configured to sample all packets matching a flow. Instance **inst1** is configured to sample one in every 10 packets.

To configure the global rate, include the **rate** statement and specify **1** as the rate at the **[edit forwarding-options sampling input]** hierarchy level. To configure the instance rate, include the **rate** statement and specify **10** as the rate at the **[edit forwarding-options sampling instance inst1 input]** hierarchy level.

```
[edit forwarding-options]
sampling {
  input {
    rate 1;
  }
  instance inst1 {
    input {
      rate 10;
    }
  }
}
```

6. Apply the sampling instance to the desired FPC or DPC.

The FPC number must match the FPC portion of the interface name for the interface on which sampling is enabled.

To apply the sampling instance, include the **sampling-instance** statement and specify **ins1** at the **[edit chassis fpc 0]** hierarchy level.

```
[edit]
chassis {
  fpc 0 {
    sampling-instance ins1;
  }
}
```

7. Configure the flow collector and enable active flow monitoring for IPv4 and for MPLS using the version 9 template format.
 - To configure the flow collector for IPv4, include the **flow-server** statement and specify the IP address of the host system that is collecting traffic flows using version 9 at the **[edit forwarding-options sampling instance ins1 family inet output]** hierarchy level. Also include the **port** statement and specify UDP port **2055** for use by the flow collector.

To enable active flow monitoring for IPv4 using the version 9 template format, include the **template** statement and specify **ipv4-template** as the name of the template to use at the **[edit forwarding-options sampling instance ins1 family inet output flow-server 100.1.1.2 version9]** hierarchy level.

- To configure the flow collector for MPLS, include the **flow-server** statement and specify the IP address of the host system that is collecting traffic flows using version 9 at the **[edit forwarding-options sampling instance ins1 family mpls output]** hierarchy level. Also include the **port** statement and specify UDP port **2055** for use by the flow collector.

To enable active flow monitoring for MPLS using the version 9 template format, include the **template** statement and specify **mpls** as the name of the template to use at the **[edit forwarding-options sampling instance ins1 family mpls output flow-server 100.1.1.2 version9]** hierarchy level.

```
[edit forwarding-options sampling instance ins1]
family inet {
  output {
    flow-server 100.1.1.2 {
      port 2055;
      version9 {
        template v4_template;
      }
    }
  }
}
family mpls {
  output {
    flow-server 100.1.1.2 {
      port 2055;
      version9 {
        template mpls;
      }
    }
  }
}
```

```

    }
  }
}

```

8. Configure the IPv4 source address for the services PIC to be used in flow export.

- To configure the IPv4 source address for the **sp-0/0/0** interface, include the **source-address** statement and specify **3.3.3.3** at the **[edit forwarding-options sampling instance ins1 family inet output interface sp-0/0/0]** hierarchy level.
- To configure the IPv4 source address for the **sp-0/0/0** interface for MPLS, include the **source-address** statement and specify **3.3.3.3** at the **[edit forwarding-options sampling instance ins1 family mpls output interface sp-0/0/0]** hierarchy level.

```

[edit forwarding-options sampling instance ins1]
family inet {
  output {
    interface sp-0/0/0 {
      source-address 3.3.3.3;
    }
  }
}
family mpls {
  output {
    interface sp-0/0/0 {
      source-address 3.3.3.3;
    }
  }
}

```

9. Configure the firewall filter.

The firewall filter identifies the traffic flows that need to be sampled and processed by the services PIC. Note that the implied “from” clause in the filter determines the packets that are matched and sampled according to the sampling rate.

- To configure the firewall filter for IPv4, include the **filter** statement and specify **ipv4_sample_filter** as the name of the filter at the **[edit firewall family inet]** hierarchy level. Include the **term** statement and specify **1** as the name of the term. For active monitoring using version 9, you must include the **sample** and **accept** action statements at the **[edit firewall family inet filter ipv4_sample_filter term 1 then]** hierarchy level.
- To configure the firewall filter for MPLS, include the **filter** statement and specify **mpls_sample_filter** as the name of the filter at the **[edit firewall family mpls]** hierarchy level. Include the **term** statement and specify **1** as the name of the term. For active monitoring using version 9, you must include the **sample** and **accept** action statements at the **[edit firewall family mpls filter mpls_sample_filter term 1 then]** hierarchy level.

```

[edit firewall]
family inet {
  filter ipv4_sample_filter {

```

```

        term 1 {
            then {
                sample;
                accept;
            }
        }
    }
}
family mpls {
    filter mpls_sample_filter {
        term 1 {
            then {
                sample;
                accept;
            }
        }
    }
}

```

10. Apply the firewall filter to the set of media interfaces where traffic flow needs to be sampled for IPv4 and MPLS.

- To apply the firewall filter to the **fe-0/3/2** interface for IPv4, include the **input** statement and specify **ipv4_sample_filter** as the name of the filter at the **[edit interfaces fe-0/3/2 unit 0 family inet filter]** hierarchy level.
- To apply the firewall filter to the **fe-0/3/2** interface for MPLS, include the **input** statement and specify **mpls_sample_filter** as the name of the filter at the **[edit interfaces fe-0/3/2 unit 0 family mpls filter]** hierarchy level.

```

[edit]
interfaces {
    fe-0/3/2 {
        unit 0 {
            family inet {
                filter {
                    input ipv4_sample_filter;
                }
            }
            family mpls {
                filter {
                    input mpls_sample_filter;
                }
            }
        }
    }
}

```

Results For your reference, the relevant sample configuration for the IPv4 and MPLS flow collector follows.

```

[edit]
services {
    flow-monitoring {
        version9 {

```

```

template mpls-ipv4 {
    flow-active-timeout 600;
    flow-inactive-timeout 30;
    mpls-ipv4-template {
        label-position [ 1 2 ];
    }
    template-refresh-rate {
        packets 480000;
        seconds 600;
    }
    option-refresh-rate {
        packets 480000;
        seconds 600;
    }
}
}
}
}
forwarding-options {
    sampling {
        input {
            rate 1;
        }
        instance ins1 {
            input {
                rate 10;
            }
            family inet {
                output {
                    flow-server 100.1.1.2 {
                        port 2055;
                        version9 {
                            template v4_template;
                        }
                    }
                }
                interface sp-0/0/0 {
                    source-address 3.3.3.3;
                }
            }
        }
    }
    family mpls {
        output {
            flow-server 100.1.1.2 {
                port 2055;
                version9 {
                    template mpls;
                }
            }
            interface sp-0/0/0 {
                source-address 3.3.3.3;
            }
        }
    }
}
}
}
}

```

```
chassis {
  fpc 0 {
    sampling-instance ins1;
  }
}
firewall {
  family inet {
    filter ipv4_sample_filter {
      term 1 {
        then {
          sample;
          accept;
        }
      }
    }
  }
  family mpls {
    filter mpls_v4_sample_filter {
      term 1 {
        then {
          sample;
          accept;
        }
      }
    }
  }
}
interfaces {
  fe-0/3/2 {
    unit 0 {
      family inet {
        filter {
          input ipv4_sample_filter;
        }
      }
      family mpls {
        filter {
          input mpls_sample_filter;
        }
      }
    }
  }
  sp-0/0/0 {
    unit 0 {
      family inet;
      family mpls;
    }
  }
  ge-1/1/3 {
    description to-flow-collector;
    unit 0 {
      family inet {
        address 100.1.1/24;
      }
    }
  }
}
```


}

**Related
Documentation**

- [Flow Monitoring Overview on page 1](#)
- [Active Flow Monitoring Overview on page 3](#)
- [Active Flow Monitoring Applications on page 5](#)
- [Best Practices for Configuring Active Flow Monitoring Version 9 on page 7](#)
- [Example: Configuring Active Flow Monitoring Version 9 for IPv4 on page 9](#)
- [Example: Configuring Active Flow Monitoring Version 9 for IPv6 on page 17](#)
- [Example: Configuring Active Flow Monitoring Version 9 for MPLS on page 25](#)
- [Example: Configuring Active Flow Monitoring Version 9 for IPv4, MPLS, and IPv6 on page 43](#)
- [Verifying Active Flow Monitoring Version 9 on page 57](#)

Example: Configuring Active Flow Monitoring Version 9 for IPv4, MPLS, and IPv6

This example shows how to monitor IPv4, MPLS, and IPv6 flows by using active flow Monitoring version 9. It is organized in the following sections:

- Requirements on page 43
- Overview of Flow Monitoring on page 43
- Configuring Active Flow Monitoring Version 9 for IPv4, MPLS, and IPv6 on page 43

Requirements

This example requires the following hardware and software components:

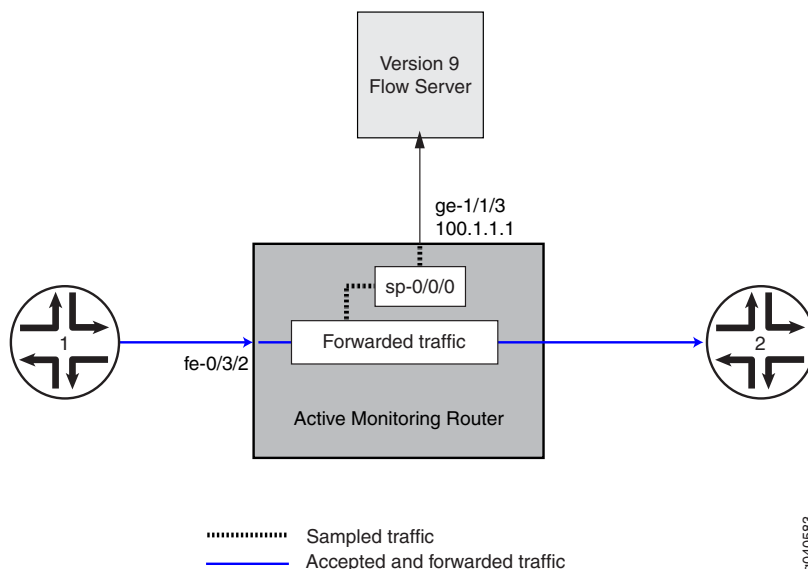
- Junos OS Release 9.2 or later
- One M 120 or M320 Multiservice Edge Router, MX Series 3D Universal Edge Router, or T Series Core Router
- One Adaptive Services PIC

Overview of Flow Monitoring

This example explains how to monitor IPv4, MPLS, and IPv6 flows.

The physical connections used in this example are shown in Figure 5 on page 43.

Figure 5: Active Flow Monitoring Version 9 for IPv4, MPLS, and IPv6 Topology



Configuring Active Flow Monitoring Version 9 for IPv4, MPLS, and IPv6

Step-by-Step Procedure

1. Enable the services PIC interface to process IPv4, MPLS, and IPv6 addresses by including the **family** statement and specifying the **inet** option, **mpls** option, and **inet6** option at the **[edit interfaces sp-0/0/0 unit 0]** hierarchy level.

```
[edit interfaces]
sp-0/0/0 {
  unit 0 {
    family inet;
    family mpls;
    family inet6;
  }
}
```

2. Configure the interface connected to the flow collector by including the **address** statement and specifying **100.1.1.1/24** as the IPv4 address of the interface at the **[edit interfaces ge-1/1/3 unit 0 family inet]** hierarchy level.

```
[edit interfaces]
ge-1/1/3 {
  description to-flow-collector;
  unit 0 {
    family inet {
      address 100.1.1.1/24;
    }
  }
}
```

3. Create the version 9 templates and configure the timers for IPv4.

Create a version 9 template for IPv4 by including the **template** statement and specifying **v4_template** as the name of the template at the **[edit services flow-monitoring version9]** hierarchy level.

Enable the template for IPv4 flows by including the **ipv4-template** statement at the **[edit services flow-monitoring version9 template v4_template]** hierarchy level.

Configure the flow active timeout by including the **flow-active-timeout** statement and specifying **600** seconds at the **[edit services flow-monitoring version9 template v4_template]** hierarchy level. Configure the flow inactive timeout by including the **flow-inactive-timeout** statement and specifying **30** seconds at the **[edit services flow-monitoring version9 template v4_template]** hierarchy level.

```
[edit services]
flow-monitoring {
  version9 {
    template v4_template {
      flow-active-timeout 600;
      flow-inactive-timeout 30;
      ipv4-template;
    }
  }
}
```

4. Create the version 9 templates and configure the timers for MPLS.

Create a version 9 template for MPLS by including the **template** statement and specifying **mpls** as the name of the template at the **[edit services flow-monitoring version9]** hierarchy level.

Enable the template for MPLS flows by including the **mpls-template** statement at the **[edit services flow-monitoring version9 template mpls]** hierarchy level. Also

include the **label-position** statement and specify label positions 1 and 2 at the **[edit services flow-monitoring version9 template mpls mpls-template]** hierarchy level.

Configure the flow active timeout by including the **flow-active-timeout** statement and specifying 600 seconds at the **[edit services flow-monitoring version9 template mpls]** hierarchy level. Configure the flow inactive timeout by including the **flow-inactive-timeout** statement and specifying 30 seconds at the **[edit services flow-monitoring version9 template mpls]** hierarchy level.

```
[edit services]
flow-monitoring {
  version9 {
    template mpls {
      flow-active-timeout 600;
      flow-inactive-timeout 30;
      mpls-template {
        label-position [ 1 2 ];
      }
    }
  }
}
```

5. Create the version 9 templates and configure the timers for IPv6.

Create a version 9 template for IPv6 by including the **template** statement and specifying **v6_template** as the name of the template at the **[edit services flow-monitoring version9]** hierarchy level.

Enable the template for IPv6 flows by including the **ipv6-template** statement at the **[edit services flow-monitoring version9 template v6_template]** hierarchy level.

Configure the flow active timeout by including the **flow-active-timeout** statement and specifying 600 seconds at the **[edit services flow-monitoring version9 template v6_template]** hierarchy level. Configure the flow inactive timeout by including the **flow-inactive-timeout** statement and specifying 30 seconds at the **[edit services flow-monitoring version9 template v6_template]** hierarchy level.

```
[edit services]
flow-monitoring {
  version9 {
    template v6_template {
      flow-active-timeout 600;
      flow-inactive-timeout 30;
      ipv6-template;
    }
  }
}
```

6. Configure the rate at which the router sends template definitions and options to the flow collector for IPv4.

Since version 9 flow monitoring traffic is unidirectional from the monitor (router) to the flow collector, configure the monitor to send template definitions and options, such as sampling rate, to the collector.

In this example, the template definitions and options are refreshed every 600 seconds or 480000 packets, whichever occurs first.

Include the **packets** statement and specify **480000** packets at the **[edit services flow-monitoring version9 template v4_template template-refresh-rate]** and **[edit services flow-monitoring version9 template v4_template option-refresh-rate]** hierarchy levels. Include the **seconds** statement and specify **600** seconds at the **[edit services flow-monitoring template v4_template version9 template-refresh-rate]** and **[edit services flow-monitoring version9 template v4_template option-refresh-rate]** hierarchy levels.

```
[edit services flow-monitoring version9 template v4_template]
template-refresh-rate {
  packets 480000;
  seconds 600;
}
option-refresh-rate {
  packets 480000;
  seconds 600;
}
```

7. Configure the rate at which the router sends template definitions and options to the flow collector for MPLS.

Include the **packets** statement and specify **480000** packets at the **[edit services flow-monitoring version9 template mpls template-refresh-rate]** and **[edit services flow-monitoring version9 template mpls option-refresh-rate]** hierarchy levels. Include the **seconds** statement and specify **600** seconds at the **[edit services flow-monitoring version9 template mpls template-refresh-rate]** and **[edit services flow-monitoring version9 template mpls option-refresh-rate]** hierarchy levels.

```
[edit services flow-monitoring version9 template mpls]
template-refresh-rate {
  packets 480000;
  seconds 600;
}
option-refresh-rate {
  packets 480000;
  seconds 600;
}
```

8. Configure the rate at which the router sends template definitions and options to the flow collector for IPv6.

Include the **packets** statement and specify **480000** packets at the **[edit services flow-monitoring version9 template v6_template template-refresh-rate]** and **[edit services flow-monitoring version9 template v6_template option-refresh-rate]** hierarchy levels. Include the **seconds** statement and specify **600** seconds at the **[edit services flow-monitoring version9 template v6_template template-refresh-rate]** and **[edit services flow-monitoring version9 template v6_template option-refresh-rate]** hierarchy levels.

```
[edit services flow-monitoring version9 template v6_template]
template-refresh-rate {
  packets 480000;
```

```

        seconds 600;
    }
    option-refresh-rate {
        packets 480000;
        seconds 600;
    }

```

9. Configure the sampling rate and run length.

The sampling rate determines the ratio of the number of packets to be sampled. For example, if you specify a rate of 10, 1 out of every 10 packets is sampled. In this example, the rate is 1 out of every 1 packets.

Sampling can be configured as a global chassis configuration that is applicable to all Flexible PIC Concentrators (FPCs) and Dense Port Concentrators (DPCs) at the **[edit forwarding-options sampling input]** hierarchy level. Sampling can also be configured at the **[edit forwarding-options sampling instance *instance-name*]** hierarchy level and then applied to a single FPC.

The run length sets the number of samples to be taken following the initial trigger event. This allows you to sample packets following those already being sampled. Since you are sampling every packet in this example, the run length can be set to 1.

To configure the sampling rate, include the **rate** statement and specify 1 as the rate at the **[edit forwarding-options sampling instance *ins1* input]** hierarchy level. To configure the run length, include the **run-length** statement and specify 1 as the run length at the **[edit forwarding-options sampling instance *ins1* input]** hierarchy level.

```

[edit forwarding-options]
sampling {
    instance ins1 {
        input {
            rate 1;
            run-length 1;
        }
    }
}

```

10. Apply the sampling instance to the desired FPC or DPC.

The FPC number must match the FPC portion of the interface name for the interface on which sampling is enabled.

To apply the sampling instance, include the **sampling-instance** statement and specify **ins1** at the **[edit chassis fpc 0]** hierarchy level.

```

[edit]
chassis {
    fpc 0 {
        sampling-instance ins1;
    }
}

```

11. Configure the flow collector and enable active flow monitoring for IPv4, MPLS, and IPv6 using the version 9 template format.

- To configure the flow collector for IPv4, include the **flow-server** statement and specify **100.1.1.2** as the IPv4 address of the host system that is collecting traffic flows using version 9 at the **[edit forwarding-options sampling instance ins1 family inet output]** hierarchy level. Also include the **port** statement and specify UDP port **2055** for use by the flow collector.

To enable active flow monitoring for IPv4 using the version 9 template format, include the **template** statement and specify the **v4-template** as the name of the template to use at the **[edit forwarding-options sampling instance ins1 family inet output flow-server 100.1.1.2 version9]** hierarchy level.

- To configure the flow collector for MPLS, include the **flow-server** statement and specify **100.1.1.2** as the IPv4 address of the host system that is collecting traffic flows using version 9 at the **[edit forwarding-options sampling instance ins1 family mpls output]** hierarchy level. Also include the **port** statement and specify UDP port **2055** for use by the flow collector.

To enable active flow monitoring for MPLS using the version 9 template format, include the **template** statement and specify **mpls** as the name of the template to use at the **[edit forwarding-options sampling instance ins1 family mpls output flow-server 100.1.1.2 version9]** hierarchy level.

- To configure the flow collector for IPv6, include the **flow-server** statement and specify **100.1.1.2** as the IPv4 address of the host system that is collecting traffic flows using version 9 at the **[edit forwarding-options sampling instance ins1 family inet6 output]** hierarchy level. Also include the **port** statement and specify UDP port **2055** for use by the flow collector.

To enable active flow monitoring using the version 9 template format, include the **template** statement and specify **v6-template** as the name of the template to use at the **[edit forwarding-options sampling instance ins1 family inet6 output flow-server 100.1.1.2 version9]** hierarchy level.

```
[edit forwarding-options sampling instance ins1]
family inet {
  output {
    flow-server 100.1.1.2 {
      port 2055;
      version9 {
        template v4-template;
      }
    }
  }
}
family mpls {
  output {
    flow-server 100.1.1.2 {
      port 2055;
      version9 {
        template mpls;
      }
    }
  }
}
```



```

    }
  }
}
family inet6 {
  output {
    flow-server 100.1.1.2 {
      port 2055;
      version9 {
        template v6_template;
      }
    }
  }
}
}

```

12. Configure the IPv4 source address for the service PIC to be used in flow export.

- To configure the IPv4 source address for the **sp-0/0/0** interface for IPv4, include the **source-address** statement and specify **3.3.3.3** at the **[edit forwarding-options sampling instance ins1 family inet output interface sp-0/0/0]** hierarchy level.
- To configure the IPv4 source address for the **sp-0/0/0** interface for MPLS, include the **source-address** statement and specify **3.3.3.3** at the **[edit forwarding-options sampling instance ins1 family mpls output interface sp-0/0/0]** hierarchy level.
- To configure the IPv4 source address for the **sp-0/0/0** interface for IPv6, include the **source-address** statement and specify **3.3.3.3** at the **[edit forwarding-options sampling instance ins1 family inet6 output interface sp-0/0/0]** hierarchy level.

```

[edit forwarding-options sampling instance ins1]
family inet {
  output {
    interface sp-0/0/0 {
      source-address 3.3.3.3;
    }
  }
}
family inet6 {
  output {
    interface sp-0/0/0 {
      source-address 3.3.3.3;
    }
  }
}
family mpls {
  output {
    interface sp-0/0/0 {
      source-address 3.3.3.3;
    }
  }
}
}

```

13. Configure the firewall filters.

The firewall filters identify the traffic flows that need to be sampled and processed by the services PIC. Note that the implied “from” clause in the filter determines the packets that are matched and sampled according to the sampling rate.

- To configure the firewall filter for IPv4, include the **filter** statement and specify **ipv4_sample_filter** as the name of the filter at the **[edit firewall family inet]** hierarchy level. Include the **term** statement and specify 1 as the name of the term. For active monitoring using version 9, you must include the **sample** and **accept** action statements at the **[edit firewall family inet filter ipv4_sample_filter term 1 then]** hierarchy level.
- To configure the firewall filter for MPLS, include the **filter** statement and specify **mpls_sample_filter** as the name of the filter at the **[edit firewall family mpls]** hierarchy level. Include the **term** statement and specify 1 as the name of the term. For active monitoring using version 9, you must include the **sample** and **accept** action statements at the **[edit firewall family mpls filter mpls_sample_filter term 1 then]** hierarchy level.
- To configure the firewall filter for IPv6, include the **filter** statement and specify **ipv6_sample_filter** as the name of the filter at the **[edit firewall family inet6]** hierarchy level. Include the **term** statement and specify 1 as the name of the term. For active monitoring using version 9, you must include the **sample** and **accept** action statements at the **[edit firewall family inet6 filter ipv6_sample_filter term 1 then]** hierarchy level.

```
[edit firewall]
family inet {
  filter ipv4_sample_filter {
    term 1 {
      then {
        sample;
        accept;
      }
    }
  }
}
family mpls {
  filter mpls_sample_filter {
    term 1 {
      then {
        sample;
        accept;
      }
    }
  }
}
family inet6 {
  filter ipv6_sample_filter {
    term 1 {
      then {
        sample;
```

```

        accept;
    }
}
}

```

14. Apply the firewall filter to the set of media interfaces where traffic flow needs to be sampled.

- To apply the firewall filter to the **fe-0/3/2** interface for IPv4, include the **input** statement and specify **ipv4_sample_filter** as the name of the filter at the **[edit interfaces fe-0/3/2 unit 0 family inet filter]** hierarchy level.
- To apply the firewall filter to the **fe-0/3/2** interface for MPLS, include the **input** statement and specify **mpls_sample_filter** as the name of the filter at the **[edit interfaces fe-0/3/2 unit 0 family mpls filter]** hierarchy level.
- To apply the firewall filter to the **fe-0/3/2** interface for IPv6, include the **output** statement and specify **ipv6_sample_filter** as the name of the filter at the **[edit interfaces fe-0/3/2 unit 0 family inet6 filter]** hierarchy level.

```

[edit]
interfaces {
  fe-0/3/2 {
    unit 0 {
      family inet {
        filter {
          input ipv4_sample_filter;
        }
      }
      family mpls {
        filter {
          input mpls_sample_filter;
        }
      }
      family inet6 {
        filter {
          input ipv6_sample_filter;
        }
      }
    }
  }
}

```

Results For your reference, the relevant sample configuration for the flow collector follows.

```

chassis {
  fpc 0 {
    sampling-instance ins1;
  }
}
interfaces {
  fe-0/3/2 {
    unit 0 {
      family inet {

```

```
        filter {
            input ipv4_sample_filter;
        }
    }
    family inet6 {
        filter {
            input ipv6_sample_filter;
        }
    }
    family mpls {
        filter {
            input mpls_sample_filter;
        }
    }
}
}
ge-1/1/3 {
    unit 0 {
        family inet {
            address 100.1.1/24;
        }
    }
}
sp-0/0/0 {
    unit 0 {
        family inet;
        family inet6;
        family mpls;
    }
}
forwarding-options {
    sampling {
        instance {
            ins1 {
                input {
                    rate 1;
                    run-length 1;
                }
                family inet {
                    output {
                        flow-server 100.1.1.2 {
                            port 2055;
                            version9 {
                                template {
                                    v4_template;
                                }
                            }
                        }
                    }
                }
                interface sp-0/0/0 {
                    source-address 3.3.3.3;
                }
            }
        }
        family inet6 {
            output {
```

```

        flow-server 100.1.1.2 {
            port 2055;
            version9 {
                template {
                    v6_template;
                }
            }
        }
        interface sp-0/0/0 {
            source-address 3.3.3.3;
        }
    }
    family mpls {
        output {
            flow-server 100.1.1.2 {
                port 2055;
                version9 {
                    template {
                        mpls;
                    }
                }
            }
        }
        interface sp-0/0/0 {
            source-address 3.3.3.3;
        }
    }
}

firewall {
    family inet {
        filter ipv4_sample_filter {
            term 1 {
                then {
                    sample;
                    accept;
                }
            }
        }
    }
    family inet6 {
        filter ipv6_sample_filter {
            term 1 {
                then {
                    sample;
                    accept;
                }
            }
        }
    }
    family mpls {
        filter mpls_sample_filter {
            term 1 {

```

Copyright © 2011, Juniper Networks, Inc.

**Related
Documentation**

- [Flow Monitoring Overview on page 1](#)
- [Active Flow Monitoring Overview on page 3](#)
- [Best Practices for Configuring Active Flow Monitoring Version 9 on page 7](#)
- [Example: Configuring Active Flow Monitoring Version 9 for IPv4 on page 9](#)
- [Example: Configuring Active Flow Monitoring Version 9 for IPv6 on page 17](#)
- [Example: Configuring Active Flow Monitoring Version 9 for MPLS on page 25](#)
- [Example: Configuring Active Flow Monitoring Version 9 for MPLS and IPv4 on page 33](#)
- [Verifying Active Flow Monitoring Version 9 on page 57](#)



NOTE: The verification steps shown for active flow monitoring are linked to multiple configuration examples and do not exactly match the configuration of any single example.

Verify the operation of active flow monitoring by doing the following:

- Verifying That Active Flow Monitoring Is Working on page 57
- Verifying That the Services PIC Is Operational for Active Flow Monitoring on page 57
- Verifying That Sampling Is Enabled and the Filter Direction Is Correct for Active Flow Monitoring on page 58
- Verifying That the Sampling Instance Is Applied to the Correct FPC for Active Flow Monitoring on page 59
- Verifying That the Route Record Is Being Created for Active Flow Monitoring on page 59
- Verifying That the Sampling Process Is Running for Active Flow Monitoring on page 60
- Verifying That the TCP Connection Is Operational for Active Flow Monitoring on page 60
- Verifying That the Services PIC Memory Is Not Overloaded for Active Flow Monitoring on page 61
- Verifying That the Active Flow Monitoring Flow Collector Is Reachable on page 61

Verifying That Active Flow Monitoring Is Working

Purpose Verify that active flow monitoring is working.

Action To verify that active flow monitoring is working, use the **show services accounting flow** command.

```
user@host> show services accounting flow
Flow information
Service Accounting interface: sp-0/0/0, Local interface index: 149
Flow packets: 87168293, Flow bytes: 5578770752
Flow packets 10-second rate: 45762, Flow bytes 10-second rate: 2928962
Active flows: 1000, Total flows: 2000
Flows exported: 19960, Flows packets exported: 582
Flows inactive timed out: 1000, Flows active timed out: 29000
```

Meaning The output shows that active flows exist and that flow packets are being exported. This indicates that flow monitoring is working. If flow monitoring is not working, verify that the services PIC is present in the chassis and is operational.

Verifying That the Services PIC Is Operational for Active Flow Monitoring

Purpose Verify that the services PIC configured for active flow monitoring is present in the chassis and is operational.

Action To verify that the services PIC is operational, use the **show chassis hardware** command.

```

user@host> show chassis hardware
Item                Version  Part number  Serial number  Description
Chassis              REV 04   710-018041   JN108DA32AEA   M120
Midplane             REV 06   710-011407   RC2209         M120 Midplane
FPM Board            REV 02   710-011407   DM3120         M120 FPM Board
FPM Display          REV 02   710-011405   DN1536         M120 FPM Display
FPM CIP              REV 05   710-011410   DK5856         M120 FPM CIP
PEM 0                Rev 04   740-011936   001830         AC Power Entry Module
Routing Engine 0     REV 07   740-014080   1000743523     RE-A-1000
Routing Engine 1     REV 07   740-014080   1000743527     RE-A-1000
CB 0                 REV 09   710-011403   DP4953         M120 Control Board
CB 1                 REV 09   710-011403   DP5107         M120 Control Board
FPC 3                REV 03   710-015835   DL6175         M120 FPC Type 1
  PIC 0              REV 12   750-003033   RF2269         4x OC-3 SONET, MM
  PIC 1              REV 13   750-012266   DL3620         4x 1GE(LAN), IQ2
    Xcvr 0           REV 01   740-013111   8154851        SFP-T
    Xcvr 1           REV 01   740-013111   8154691        SFP-T
    Xcvr 2           REV 01   740-013111   8142743        SFP-T
    Xcvr 3           REV 01   740-013111   8142607        SFP-T
  PIC 2              REV 11   750-005727   RH2029         2x OC-3 ATM-II IQ, MM
  PIC 3              REV 14   750-002911   RH0523         4x F/E, 100 BASE-TX
  Board B            REV 03   710-017980   DN2163         M120 FPC Mezz Board
FPC 4                REV 02   710-015835   DN1923         M120 FPC Type 1
  PIC 0              REV 12   750-014884   DH2850         MultiServices 100
  PIC 1              REV 13   750-014884   DZ9927         MultiServices 100
  PIC 2              REV 13   750-023755   XN9363         4x CHOC3 SONET CE SFP
    Xcvr 0           REV 01   740-012434   6455242        SFP-SR
~
~
~
  Board B            REV 03   710-017980   DN2155         M120 FPC Mezz Board
FEB 3                REV 06   710-015795   DN8222         M120 FEB
FEB 4                REV 06   710-015795   DP2649         M120 FEB
Fan Tray 0           Front Top Fan Tray
Fan Tray 1           Front Bottom Fan Tray

```

Meaning The output shows that **PIC 0** under **FPC 4** is a **MultiServices** PIC that has completed booting and is operational. If the PIC is operational but flow monitoring is not working, verify that sampling is enabled on the media interface on which traffic flow is expected and that the sampling filter direction is correct.

Verifying That Sampling Is Enabled and the Filter Direction Is Correct for Active Flow Monitoring

Purpose Verify that sampling is enabled on the media interface on which traffic flow is expected and that the sampling filter direction is correct.

Action To verify that sampling is enabled on the media interface on which traffic flow is expected and that the sampling filter direction is correct, use the **show interfaces *interface-name* extensive | grep filters** command.

```

user@host> show interfaces fe-3/3/2 extensive | grep filters
CAM destination filters: 4, CAM source filters: 0
  Input Filters: ipv4_sample_filter
  Input Filters: ipv6_sample_filter
  Input Filters: mpls_sample_filter

```

Meaning The command output shows that the sample filter is applied to the media interface on which traffic flow is expected (**fe-3/3/2**) and that the sampling filter direction is **Input**. If the PIC is operational and the filters are correct but flow monitoring is not working, verify that the sampling instance is applied to the FPC where the media interface resides.



TIP: If a firewall filter is used to enable sampling, add a counter as an action in the firewall filter. Then, verify if the counter is incrementing. If the counter is incrementing, it confirms that the traffic is present and that the filter direction is correct.

Verifying That the Sampling Instance Is Applied to the Correct FPC for Active Flow Monitoring

Purpose Verify that the sampling instance is applied to the FPC where the media interface resides.

Action To verify that the sampling instance is applied to the correct FPC, use the **show configuration chassis** command.

```
user@host> show configuration chassis
```

```
chassis {
  fpc 4 {
    sampling-instance ins1;
  }
}
```

Meaning The output shows that the sampling instance is applied to the correct FPC. If the PIC is operational, the filters are correct, and the sampling instance is applied to the correct FPC but flow monitoring is not working, verify that the route record set of data is being created.

Verifying That the Route Record Is Being Created for Active Flow Monitoring

Purpose Verify that the route record set of data is being created.

Action To verify that the route record set of data is being created, use the **show services accounting status** command.

```
user@host> show services accounting status
Service Accounting interface: sp-4/0/0
Export format: 9, Route record count: 40
IFL to SNMP index count: 11, AS count: 1
Configuration set: Yes, Route record set: Yes, IFL SNMP map set: Yes
Route record set: Yes, IFL SNMP map set: Yes
```

Meaning The output shows that the **Route record set** field is set to **Yes**. This confirms that the route record set is created.



TIP: If the route record set field is set to no, the record might not have been downloaded yet. Wait for 60-100 seconds and check again. If the route record is still not created, verify that the sampling process is running, that the connection between the PIC and the process is operational, and that the PIC memory is not overloaded.

Verifying That the Sampling Process Is Running for Active Flow Monitoring

Purpose Verify that the sampling process is running.

Action To verify that the sampling process is running, use the **show system processes extensive | grep sampled** command.

```
user@host> show system processes extensive | grep sampled
PID USERNAME  THR PRI NICE   SIZE  RES  STATE  TIME  WCPU  COMMAND
1581 root        1   1  111   5660K 5108K select   0:00  0.00% sampled
```

Meaning The output shows that **sampled** is listed as a running system process. In addition to verifying that the process is running, verify that the TCP connection between the sampled process and the services PIC is operational.

Verifying That the TCP Connection Is Operational for Active Flow Monitoring

Purpose Verify that the TCP connection between the sampled process and the services PIC is operational.

Action To verify that the TCP connection is operational, use the **show system connections inet | grep 6153** command.

```
user@host> show system connections inet | grep 6153
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         (state)
~
~
~
tcp        0      0 128.0.0.1.6153          128.0.2.17.11265       ESTABLISHED
tcp4       0      0 *.6153                  *.*                     LISTEN
```

Meaning The output shows that the TCP connection between the sampled process socket (**6153**) and the services PIC (**128.0.0.1**) is **ESTABLISHED**. In addition to verifying that the TCP connection between the sampled process and the services PIC is operational, verify that the services PIC memory is not overloaded.



TIP: If the TCP connection between the sampled process and the services PIC is not established, restart the sampled process by using the **restart sampling** command.

Verifying That the Services PIC Memory Is Not Overloaded for Active Flow Monitoring

Purpose Verify that the services PIC memory is not overloaded.

Action To verify that the services PIC memory is not overloaded, use the **show services accounting errors** command.

```
user@host> show services accounting errors
Service Accounting interface: sp-4/0/0, Local interface index: 542
Service name: (default sampling)
~
~
~ Error information
  Service sets dropped: 0, Active timeout failures: 0
  Export packet failures: 0, Flow creation failures: 0
  Memory overload: No
```

Meaning The output shows that the memory overload field is set to **No**, indicating that the PIC memory is not overloaded. As a final check that active flow monitoring is working, verify that the flow collector is reachable.

Verifying That the Active Flow Monitoring Flow Collector Is Reachable

Purpose Verify that flow collector is reachable by using the **ping** command.

Action From the router, issue the **ping** command to the flow collector.

```
user@host> ping 100.1.1.2
PING 100.1.1.2 (100.1.1.2): 56 data bytes
64 bytes from 100.1.1.2: icmp_seq=0 ttl=64 time=0.861 ms
64 bytes from 100.1.1.2: icmp_seq=1 ttl=64 time=0.869 ms
64 bytes from 100.1.1.2: icmp_seq=2 ttl=64 time=0.786 ms
^C
--- 4.4.4.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.786/0.839/0.869/0.037 ms
```

Meaning The output shows **0% packet loss** indicating that the flow collector can be reached.



TIP: Verify that the flow collector is reachable through the media interface and is not being reached through the fxp0 Ethernet management interface.

Related Documentation

- Flow Monitoring Overview on page 1
- Active Flow Monitoring Overview on page 3
- Active Flow Monitoring Applications on page 5
- Best Practices for Configuring Active Flow Monitoring Version 9 on page 7
- Example: Configuring Active Flow Monitoring Version 9 for IPv4 on page 9

- Example: Configuring Active Flow Monitoring Version 9 for IPv6 on page 17
- Example: Configuring Active Flow Monitoring Version 9 for MPLS on page 25
- Example: Configuring Active Flow Monitoring Version 9 for MPLS and IPv4 on page 33
- Example: Configuring Active Flow Monitoring Version 9 for IPv4, MPLS, and IPv6 on page 43