



# NetFlow Input Filters

---

Full NetFlow accounts for all traffic ingress to the subinterface on which it is enabled. But in some cases, you must gather NetFlow data only on a specific subset of traffic. The NetFlow Input Filters feature provides this capability by letting you create filters to select flows for NetFlow processing. For example, you can select flows from a specific group of hosts. This feature also lets you select various sampling rates for selected flows.

This feature requires the classification of packets in a variety of ways: IP source and destination addresses, Layer 4 protocol and port numbers, incoming interface, MAC address, IP Precedence, DSCP value, Layer 2 information (such as Frame-Relay DE bits or Ethernet 802.1p bits), and NBAR (Network-Based Application Recognition) information. The packets are classified (filtered) on the above criteria, and flow accounting is applied to them on subinterfaces.

This feature is closely related to the Random Sampled NetFlow features, because both features provide ways to limit incoming traffic to only traffic of interest for NetFlow processing. The Random Sampled NetFlow algorithms are applied after input filtering.

## History for NetFlow Input Filters Feature

| Release     | Modification  |
|-------------|---|
| 12.3(4)T    | This feature was introduced.                                    |
| 12.2(25)S   | This feature was integrated into Cisco IOS Release 12.2(25)S.   |
| 12.2(27)SBC | This feature was integrated into Cisco IOS Release 12.2(27)SBC. |

## Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

## Contents

- [Prerequisites for NetFlow Input Filters, page 2](#)
- [Restrictions for NetFlow Input Filters, page 2](#)



---

**Corporate Headquarters:**  
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2003–2005 Cisco Systems, Inc. All rights reserved.

- [Information About NetFlow Input Filters, page 2](#)
- [How to Configure NetFlow Input Filters, page 4](#)
- [Configuration Examples for NetFlow Input Filters, page 6](#)
- [Additional References, page 8](#)
- [Command Reference, page 10](#)
- [Glossary, page 16](#)

## Prerequisites for NetFlow Input Filters

Before you can configure the NetFlow Input Filters feature, you must configure CEF switching or dCEF switching, fast switching is not supported.

## Restrictions for NetFlow Input Filters

On Cisco 7500 series platforms, the NetFlow Input Filters feature is supported only in distributed mode.

## Information About NetFlow Input Filters

To configure the NetFlow Input Filters feature, you must understand the following concepts:

- [Flow, page 2](#)
- [Modular QoS Command-Line Interface \(MQC\), page 3](#)
- [Filtering, page 3](#)
- [Export Format, page 3](#)
- [Subinterface Support, page 3](#)
- [Memory Impact, page 3](#)
- [Performance Impact, page 3](#)

## Flow

NetFlow provides highly granular per-flow traffic statistics in a Cisco router. A flow is a unidirectional set of packets that arrive at the router on the same subinterface, have the same source and destination IP addresses, Layer 4 protocol, TCP/UDP source and destination ports, and the same type of service (TOS) byte in the IP headers. The router accumulates NetFlow statistics in a NetFlow cache and can export them to an external device (such as the Cisco CNS NetFlow Collection Engine) for further processing.

## Modular QoS Command-Line Interface (MQC)

The filtering mechanism uses the Modular QoS Command-Line Interface (MQC) to classify flows. You can create multiple filters with matching samplers on a per subinterface basis. For example, you can subdivide subinterface traffic into multiple classes based on ToS values or destination prefix (or both), and for each class, configure sampling at a different rate—using higher rates for higher-priority classes of traffic and lower rates for lower-priority ones.

Example uses of NetFlow Input Filters are class-based traffic analysis and monitoring on-network or off-network traffic.

## Filtering

MQC has many policies (actions) such as bandwidth and queuing. These policies are applied only if a packet matches a criterion in a class map that is applied to the subinterface. A class map contains a set of “match” clauses and instructions on how to evaluate the clauses and acts as a filter for the policies, which are applied only if a packet’s content satisfies the match clause. The NetFlow Input Filters feature adds NetFlow accounting to the MQC infrastructure, which means that flow accounting is done on a packet only if it satisfies the match clauses.

There are two types of filter algorithms available: ACL-based flow-mask filters and fields of filter (source IP address, destination IP address, source application port, destination application port, port protocol, ToS bits, and TCP flags).

## Export Format

The NetFlow Input Filters feature is supported in the version 5 and version 9 NetFlow export formats.

## Subinterface Support

NetFlow Input Filters is supported at the subinterface level. You can configure NetFlow Input Filters per subinterface as well as per physical interface. You can simultaneously select more than one filter per subinterface.

## Memory Impact

This feature requires no additional memory. This feature allows a smaller NetFlow cache than full NetFlow, because it significantly reduces the number of flows. This feature requires an insignificant amount of memory for each configured NetFlow sampler.

## Performance Impact

Accounting of classified traffic saves router resources by reducing the number of flows being processed and exported. This also conserves bandwidth depending on the usage and the class-map criteria. However, performance might degrade depending on the number and complexity of class maps configured in a policy.

# How to Configure NetFlow Input Filters

NetFlow Input Filters uses MQC, which is a framework that separates the specification of a classification policy (meaning the definition of traffic classes) and the specification of actions (such as with bandwidth management) that occur based on the results of the applied classification. This feature introduces new type of actions: sending traffic through a sampler and processing it by NetFlow if it is selected by the sampler.

This section describes how to configure a NetFlow policy and consists of the following configuration tasks:

- [Prerequisites, page 4](#) (required)
- [Creating a Policy Containing NetFlow Sampling Actions, page 4](#) (required)
- [Applying a Policy to an Interface, page 5](#) (required)
- [Troubleshooting Tips, page 6](#) (optional)

## Prerequisites

Before you configure NetFlow Input Filters, you must first create traffic classes and define NetFlow sampler maps. For more information about creating traffic classes, refer to the “Modular Quality of Service Command-Line Interface” part in the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.3 and the *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.3 T. For more information about defining NetFlow sampler maps, see the *Random Sampled NetFlow* feature module, release 12.3 T.

## Creating a Policy Containing NetFlow Sampling Actions

This section shows how to create a class-based policy that contains NetFlow sampling actions. You can assign only one NetFlow Input Filters sampler to a class (assigning a subsequent NetFlow Input Filters sampler to a class overwrites the previous sampler). Removing a NetFlow sampler map also removes the NetFlow Input Filters sampler under the corresponding policy map.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** { *class-name* | **class-default** }
5. **netflow-sampler** *sampler-map-name*
6. **end**

## DETAILED STEPS

|        | Command or Action  | Purpose   |
|--------|--|---|
| Step 1 | <b>enable</b><br><br><b>Example:</b><br>Router> enable   | Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>  |
| Step 2 | <b>configure terminal</b><br><br><b>Example:</b><br>Router# configure terminal   | Enters global configuration mode.   |
| Step 3 | <b>policy-map</b> <i>policy-map-name</i><br><br><b>Example:</b><br>Router(config)# policy-map mypolicymap                            | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.  |
| Step 4 | <b>class</b> { <i>class-name</i>   <b>class-default</b> }<br><br><b>Example:</b><br>Router(config-pmap)# class high_importance_class | Specifies the name of the class whose policy you want to create or change or specifies the default class (commonly known as the class-default class) before you configure its policy. |
| Step 5 | <b>netflow-sampler</b> <i>sampler-map-name</i><br><br><b>Example:</b><br>Router(config-pmap-c)# netflow-sampler high_sampling        | Enables a NetFlow input filters sampler.  |
| Step 6 | <b>end</b><br><br><b>Example:</b><br>Router(config-pmap-c)# end  | Ends the configuration session and returns to privileged EXEC mode.   |

## Applying a Policy to an Interface

This section shows how to apply a policy containing NetFlow sampling actions to an interface. After you define a service policy with the **policy-map** command, you use the **service-policy** command in interface configuration mode to attach it to one or more interfaces to specify the service policy for those interfaces. Although you can assign the same service policy to multiple interfaces, each interface can have only one service policy attached. You can apply the service policy only in the input direction.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type-number*
4. **service-policy input** *policy-map-name*
5. **end**

## DETAILED STEPS

|        | Command or Action   | Purpose   |
|--------|---|---|
| Step 1 | <b>enable</b><br><br><b>Example:</b><br>Router> enable  | Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>  |
| Step 2 | <b>configure terminal</b><br><br><b>Example:</b><br>Router# configure terminal  | Enters global configuration mode.   |
| Step 3 | <b>interface</b> <i>type-number</i><br><br><b>Example:</b><br>Router(config)# interface POS1/0                                      | Enters interface configuration mode.  |
| Step 4 | <b>service-policy input</b> <i>policy-map-name</i><br><br><b>Example:</b><br>Router(config-if)# service-policy input<br>mypolicymap | Attaches a policy map to an input interface or virtual circuit (VC), or an output interface or VC, to be used as the service policy for that interface or VC. |
| Step 5 | <b>end</b><br><br><b>Example:</b><br>Router(config-if)# end   | Ends the configuration session and returns to privileged EXEC mode.   |

## Troubleshooting Tips

Use the **debug flow-sampler class-based** command to display debugging output for the NetFlow Input Filters feature.

## Configuration Examples for NetFlow Input Filters

This section provides the following configuration examples:

- [Creating a Policy Containing NetFlow Sampling Actions: Example, page 7](#)
- [Applying a Policy to an Interface: Example, page 7](#)

## Creating a Policy Containing NetFlow Sampling Actions: Example

The following example shows how to create a class-based policy containing NetFlow sampling actions. In this example, a sampling action named `my_high_sampling` is applied to a class named `my_high_importance_class`, a sampling action named `my_medium_sampling` is applied to a class named `my_medium_importance_class`, and a sampling action named `my_low_sampling` is applied to the default class.

```
Router> enable
Password:
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# policy-map mypolicymap
  Router(config-pmap)# class my_high_importance_class
  Router(config-pmap-c)# netflow-sampler my_high_sampling
  Router(config-pmap-c)# exit
  Router(config-pmap)# class my_medium_importance_class
  Router(config-pmap-c)# netflow-sampler my_medium_sampling
  Router(config-pmap-c)# exit
  Router(config-pmap)# class class-default
  Router(config-pmap-c)# netflow-sampler my_low_sampling
  Router(config-pmap-c)# end
Router#
3w5d:%SYS-5-CONFIG_I: Configured from console by console
```

## Applying a Policy to an Interface: Example

The following example shows how to apply a policy containing NetFlow sampling actions to an interface. In this example, a policy named `mypolicymap` is attached to interface `POS1/0` and also to interface `ATM2/0`:

```
Router> enable
Password:
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface POS1/0
  Router(config-if)# service-policy input mypolicymap
  Router(config-if)# exit
Router(config)# interface ATM2/0
  Router(config-if)# service-policy input mypolicymap
  Router(config-if)# end
Router#
3w5d:%SYS-5-CONFIG_I: Configured from console by console

Router(config)# class-map my_high_importance_class
Router(config-cmap)# match access-group 101
Router(config-cmap)# exit
Router(config)# class-map my_medium_importance_class
Router(config-cmap)# match access-group 102
Router(config-cmap)# exit

Router(config)# flow-sampler-map my_high_sampling
Router(config-sampler-map)# mode random one-out-of 1
Router(config-sampler-map)# exit
Router(config)# flow-sampler-map my_medium_sampling
Router(config-sampler-map)# mode random one-out-of 100
Router(config-sampler-map)# exit
Router(config)# flow-sampler-map my_low_sampling
```

```
Router(config-sampler-map) # mode random one-out-of 1000
Router(config-sampler-map) # exit
```

## Additional References

The following sections provide references related to NetFlow input filters:

## Related Documents

| Related Topic   | Document Title   |
|---|--|
| NetFlow   | <i>Cisco IOS Switching Services Configuration Guide</i> , Release 12.3<br><i>Cisco IOS Switching Services Command Reference</i> , Release 12.3T<br><i>Cisco IOS Command Reference Master Index</i> , Release 12.3  |
| NetFlow version 9 data export   | <i>NetFlow v9 Export Format</i> feature module, Release 12.3   |
| NetFlow version 9 export format   | <i>NetFlow Version 9 Flow-Record Format</i> white paper  |
| Description of an actual customer deployment of NetFlow services within an IP network   | <i>NetFlow Services for an Enterprise Network</i> integrated solutions document (ISD)  |
| Using MQC to create traffic classes, create class-based traffic policies, and attach class-based traffic policies to interfaces | “Modular Quality of Service Command-Line Interface” part in the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> , Release 12.3<br><i>Cisco IOS Quality of Service Solutions Command Reference</i> , Release 12.3T                      |
| IP multicast routing  | “IP Multicast” part in the <i>Cisco IOS IP Configuration Guide</i> , Release 12.3  |
| NetFlow Minimum Prefix Mask for Router-Based Aggregation feature  | <i>NetFlow Minimum Prefix Mask for Router-Based Aggregation</i> feature module, Release 12.1(3)T   |
| NetFlow ToS-Based Router Aggregation feature  | <i>NetFlow ToS-Based Router Aggregation</i> feature module, Release 12.1(3)T   |
| Sampled NetFlow feature   | <i>Sampled NetFlow</i> feature module, Release 12.0(26)S   |
| Random Sampled NetFlow feature  | <i>Random Sampled NetFlow</i> feature module, Release 12.3(2)T   |
| Cisco CNS NetFlow Collection Engine (formerly called NetFlow FlowCollector)   | <i>Cisco CNS NetFlow Collection Engine Installation and User Guide</i> , Release 4.0<br><i>Documentation Updates for Cisco CNS NetFlow Collection Engine</i> , Release 4.0<br><i>Release Notes for Cisco CNS NetFlow Collection Engine</i> , Release 4.0 |
| NetFlow Data Analyzer (formerly called NetFlow FlowAnalyzer)  | <i>Network Data Analyzer Installation and User Guide</i> , Release 3.6<br><i>Release Notes for Cisco Network Data Analyzer</i> , Release 3.6(1)  |
| NetFlow performance test results  | <i>NetFlow Performance Analysis</i> white paper  |

## Standards

| Standards | Title |
|-----------|-------|
| None      | —     |

## MIBs

| MIBs | MIBs Link  |
|------|--|
| None | To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:<br><a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a> |

## RFCs

| RFCs | Title |
|------|-------|
| None | —     |

## Technical Assistance

| Description  | Link  |
|--|---|
| Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content. | <a href="http://www.cisco.com/public/support/tac/home.shtml">http://www.cisco.com/public/support/tac/home.shtml</a> |

# Command Reference

This section documents the following new and modified commands only.

## New Commands

- [netflow-sampler](#)

## Modified Commands

- [debug flow-sampler](#)

# debug flow-sampler

To enable debugging output for NetFlow sampler activity, use the **debug flow-sampler** command in privileged EXEC mode. To disable debugging output for NetFlow sampler activity, use the **no** form of this command.

**debug flow-sampler** { **class-based** | **events** | **ipc** | **match** }

**no debug flow-sampler** { **class-based** | **events** | **ipc** | **match** }

## Syntax Description

|                    |   |
|--------------------|---|
| <b>class-based</b> | Displays debug messages for class-based NetFlow samplers.   |
| <b>events</b>      | Displays debug messages when a NetFlow sampler map is added, deleted, or applied to an interface.                                 |
| <b>ipc</b>         | Displays NetFlow sampler-related debug messages for interprocess communications (IPC) between the route processor and line cards. |
| <b>match</b>       | Displays debug messages when a packet is sampled (is matched with a NetFlow sampler).   |

## Defaults

Debugging output for NetFlow sampler activity is disabled.

## Command Modes

Privileged EXEC

## Command History

| Release     | Modification  |
|-------------|---|
| 12.3(2)T    | This command was introduced.                                    |
| 12.2(18)S   | This command was integrated into Cisco IOS Release 12.2(18)S.   |
| 12.0(26)S   | This command was integrated into Cisco IOS Release 12.0(26)S.   |
| 12.3(4)T    | The <b>class-based</b> keyword was added.                       |
| 12.2(27)SBC | This command was integrated into Cisco IOS Release 12.2(27)SBC. |

## Usage Guidelines

Because debugging output is assigned high priority in the CPU process, you should use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, you should use debug commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use.

## Examples

The following is sample output from the **debug flow-sampler events** command:

```
Router# debug flow-sampler events

Flow sampler events debugging is on
Router# configure terminal
Router(config)# no flow-sampler mysampler2
Router(config)#
```

```
5d00h: Flow: Sampler mysampler2 detached from FastEthernet0/1
5d00h: Flow: Sampler mysampler2 deleted
```

The following is sample output from the **debug flow-sampler match** command:

```
Router# debug flow-sampler match

Flow sampler match debugging is on
Router#
4d23h: Flow: Packet matched sampler mysampler1 on interface FastEthernet0/0
Router#
4d23h: Flow: Packet matched sampler mysampler1 on interface FastEthernet0/0
Router#
4d23h: Flow: Packet matched sampler mysampler1 on interface FastEthernet0/0
Router#
4d23h: Flow: Packet matched sampler mysampler1 on interface FastEthernet0/0
```

[Table 1](#) describes the significant fields shown in the display.

**Table 1** *debug flow-sampler Field Descriptions*

| Field                | Description  |
|----------------------|--|
| Sampler              | Name of the NetFlow sampler.                                 |
| id                   | Unique ID of the NetFlow sampler.                            |
| packets matched      | Number of packets matched (sampled) for the NetFlow sampler. |
| mode                 | NetFlow sampling mode.                                       |
| sampling interval is | NetFlow sampling interval (in packets).                      |

#### Related Commands

| Command                        | Description   |
|--------------------------------|---|
| <b>flow-sampler</b>            | Enables a Random Sampled NetFlow sampler.   |
| <b>flow-sampler-map</b>        | Defines a Random Sampled NetFlow sampler map.   |
| <b>ip flow-export</b>          | Enables the export of NetFlow data to a collector. (To enable version 9 data export, use <b>ip flow-export version 9</b> ; to export NetFlow sampler templates, use <b>ip flow-export template options sampler</b> .) |
| <b>mode (flow sampler map)</b> | Specifies a Random Sampled NetFlow sampling mode and sampling rate.   |
| <b>netflow-sampler</b>         | Enables a class-based NetFlow sampler.  |
| <b>show flow-sampler</b>       | Displays attributes (including mode, sampling rate, and number of sampled packets) of one or all Random Sampled NetFlow samplers.   |
| <b>show ip flow export</b>     | Displays the statistics for the NetFlow data export.  |

# netflow-sampler

To enable NetFlow accounting with input filter sampling, use the **netflow-sampler** command in QoS policy-map class configuration mode. To disable NetFlow accounting with input filter sampling, use the **no** form of this command.

**netflow-sampler** *sampler-map-name*

**no netflow-sampler** *sampler-map-name*

## Syntax Description

*sampler-map-name* Name of the NetFlow sampler map to apply to the class.

## Defaults

NetFlow accounting with input filter sampling is disabled.

## Command Modes

QoS policy-map class configuration

## Command History

| Release     | Modification  |
|-------------|---|
| 12.3(4)T    | This command was introduced.                                    |
| 12.2(27)SBC | This command was integrated into Cisco IOS Release 12.2(27)SBC. |

## Usage Guidelines

NetFlow accounting with input filter sampling cannot be run concurrently with (ingress) NetFlow accounting, egress NetFlow accounting, or random sampled NetFlow on the same interface, or subinterface. In order to run NetFlow accounting with input filter sampling, you must first disable (ingress) NetFlow accounting, egress NetFlow accounting, or random sampled NetFlow.

You can assign only one NetFlow input filter sampler to a class. Assigning another NetFlow input filter sampler to a class overwrites the previous one.

Samplers, also known as filters, are based on classes, but they are enabled on interfaces. You assign a NetFlow input filters sampler to a class by using the **netflow-sampler** command in QoS policy-map class configuration. You use the **service-policy** command to attach the policy map you defined to one or more interfaces.



### Tip

If your router is running Cisco IOS release 12.2(14)S or a later release, or Cisco IOS Release 12.2(15)T or a later release, NetFlow accounting might be enabled through the use of the **ip flow ingress** command instead of the **ip route-cache flow** command. If your router has NetFlow accounting enabled through the use of **ip flow ingress** command you must disable NetFlow accounting, using the **no** form of this command, before you apply a random sampler map for random sampled NetFlow accounting on an interface otherwise the full, un-sampled traffic will continue to be seen.

You must enable either Cisco Express Forwarding (CEF) or distributed CEF (dCEF) before using this command.

**Examples**

The following example shows how to enable NetFlow accounting with input filter sampling for one class of traffic (traffic with 10 as the first octet of the IP source address):

```
Router(config)# ip cef
Router(config)# flow-sampler-map network-10
Router(config-sampler)# mode random one-out-of 100
Router(config-sampler)# exit
Router(config)# class-map match-any network-10
Router(config-cmap)# match access-group 100
Router(config-cmap)# exit
Router(config)# policy-map network-10
Router(config-pmap)# class network-10
Router(config-pmap-c)# netflow-sampler network-10
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface Ethernet0/0
Router(config-if)# no ip route-cache flow
Router(config-if)# ip route-cache cef
Router(config-if)# interface ethernet 0/0.1
Router(config-if)# service-policy input network-10
Router(config-if)# exit
Router(config)# access-list 100 permit ip 10.0.0.0 0.255.255.255 any
```

The following output from the **show flow-sampler** command verifies that the NetFlow accounting with input filter sampling is active:

```
Router# show flow-sampler

Sampler : network-10, id : 1, packets matched : 546, mode : random sampling mode
sampling interval is : 100
```

The following output from the **show ip cache verbose flow** command shows that combination of the **access-list 100 permit ip 10.0.0.0 0.255.255.255 any** command and the **match access-group 100** command has filtered out any traffic in which the source IP address does not have 10 as the first octet:

```
Router# show ip cache verbose flow
IP packet size distribution (116 total packets):
  1-32  64  96 128 160 192 224 256 288 320 352 384 416 448 480
  .000 .155 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

  512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
  .000 .000 .000 .258 .586 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 278544 bytes
  7 active, 4089 inactive, 66 added
  3768 ager polls, 0 flow alloc failures
  Active flows timeout in 1 minutes
  Inactive flows timeout in 120 seconds
IP Sub Flow Cache, 21640 bytes
  6 active, 1018 inactive, 130 added, 62 added to flow
  0 alloc failures, 0 force free
  1 chunk, 1 chunk added
  last clearing of statistics never

Protocol      Total    Flows    Packets Bytes    Packets Active(Sec) Idle(Sec)
-----      Flows   /Sec     /Flow  /Pkt    /Sec    /Flow   /Flow
TCP-Telnet    6        0.0      1      940     0.0     8.8    51.6
TCP-FTP       5        0.0      1      640     0.0     6.9    53.4
TCP-SMTP      2        0.0      3     1040    0.0    41.7   18.5
TCP-other     36       0.0      1     1105    0.0    18.8   41.5
UDP-other     6        0.0      3      52     0.0    54.8    5.5
ICMP          4        0.0      1     628    0.0    11.3   48.8
Total:       59       0.0      1     853    0.1    20.7   39.6
```

```

SrcIf          SrcIPAddress  DstIf          DstIPAddress  Pr TOS Flgs Pkts
Port Msk AS    Port Msk AS    NextHop        B/Pk Active
Et0/0.1        10.10.10.3     Et1/0.1        172.16.10.3   06 80 00    1
0016 /0 0      0016 /0 0      0.0.0.0        840    0.0
Sampler: 1 Class: 1
Et0/0.1        10.10.10.3     Et1/0.1*       172.16.10.3   06 80 00    1
0016 /0 0      0016 /0 0      0.0.0.0        840    0.0
Sampler: 1 Class: 1 FFlags: 01
Et0/0.1        10.10.11.3     Et1/0.1        172.16.10.7   06 80 00    1
0041 /0 0      0041 /0 0      0.0.0.0        1140   0.0
Sampler: 1 Class: 1
Et0/0.1        10.10.11.1     Et1/0.1        172.16.10.5   06 80 00    3
0019 /0 0      0019 /0 0      0.0.0.0        1040   36.7
Sampler: 1 Class: 1
Et0/0.1        10.10.11.1     Et1/0.1*       172.16.10.5   06 80 00    1
0019 /0 0      0019 /0 0      0.0.0.0        1040   0.0
Sampler: 1 Class: 1 FFlags: 01
Et0/0.1        10.1.1.2       Et1/0.1        172.16.10.10  06 80 00    2
0041 /0 0      0041 /0 0      0.0.0.0        1140   37.8
Sampler: 1 Class: 1
Et0/0.1        10.10.10.1     Et1/0.1        172.16.10.1   01 80 10    1
0000 /0 0      0000 /0 0      0.0.0.0        628    0.0
Sampler: 1 Class: 1

```

**Related Commands**

| Command                                  | Description   |
|--|---|
| <b>flow-sampler</b>                      | Applies a flow sampler map for random sampled NetFlow accounting to an interface.   |
| <b>flow-sampler-map</b>                  | Defines a flow sampler map for random sampled NetFlow accounting.   |
| <b>mode (flow sampler configuration)</b> | Specifies a packet interval for NetFlow accounting random sampling mode and enables the flow sampler map.                             |
| <b>class-map</b>                         | Creates a class map to be used for matching packets to a specified class.   |
| <b>policy-map</b>                        | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy                           |
| <b>service-policy</b>                    | Attaches a policy map to an input interface or virtual circuit (VC).  |
| <b>show flow-sampler</b>                 | Displays the status of random sampled NetFlow (including mode, packet interval, and number of packets matched for each flow sampler). |
| <b>show ip cache flow</b>                | Displays a summary of the NetFlow accounting statistics.  |
| <b>show ip cache verbose flow</b>        | Displays a detailed summary of the NetFlow accounting statistics.   |
| <b>show ip flow interface</b>            | Displays NetFlow accounting configuration for interfaces.   |

# Glossary

**ACL**—Access control list. A roster of users and groups of users kept by a router to control access to or from the router for a number of services.

**BGP**—Border Gateway Protocol. Interdomain routing protocol that replaces Exterior Border Gateway Protocol (EBGP). BGP exchanges reachability information with other BGP systems. It is defined by RFC 1163.

**BGP next hop**—IP address of the next hop to be used to reach a certain destination.

**CEF**—Cisco Express Forwarding. Layer 3 IP switching technology that optimizes network performance and scalability for networks with large and dynamic traffic patterns.

**data flowset**—Collection of one or more data records that are grouped together in an export packet.

**data record**—Provides information about an IP flow that exists on the device that produced an export packet. Each group of data records (meaning each data flowset) references a previously transmitted template ID, which can be used to parse the data within the records.

**dCEF**—Distributed Cisco Express Forwarding. Type of CEF switching in which line cards (such as VIP line cards) maintain an identical copy of the forwarding information base (FIB) and adjacency tables. The line cards perform the express forwarding between port adapters; this relieves the route/switch processor of involvement in the switching operation.

**export packet**—Type of packet built by a device (for example, a router) with NetFlow services enabled that is addressed to another device (for example, a NetFlow Collection Engine). The packet contains NetFlow statistics. The other device processes the packet (parses, aggregates, and stores information on IP flows).

**fast switching**—Cisco feature in which a route cache is used to expedite packet switching through a router.

**flow**—Unidirectional stream of packets between a given source and destination—both defined by a network-layer IP address and transport-layer source and destination port numbers.

**flowset**—Collection of flow records that follow the packet header in an export packet. A flowset contains information that must be parsed and interpreted by the NetFlow Collection Engine device. There are two different types of flowsets: template flowsets and data flowsets. An export packet contains one or more flowsets, and both template and data flowsets can be mixed in the same export packet.

**MQC**—Modular QoS CLI (command line interface). A CLI structure that lets you create traffic polices and attach them to interfaces. A traffic policy contains a traffic class and one or more QoS features. A traffic class is used to classify traffic, and the QoS features in the traffic policy determine how to treat the classified traffic.

**NBAR**—Network-Based Application Recognition. A classification engine in Cisco IOS software that recognizes a wide variety of applications, including Web-based applications and client/server applications that dynamically assign Transmission Control Protocol (TCP) or UDP port numbers. After the application is recognized, the network can invoke specific services for that application. NBAR is a key part of the Cisco Content Networking architecture and works with QoS features to let you use network bandwidth efficiently.

**NetFlow**—Cisco IOS acceleration and accounting feature that maintains per-flow information.

**NetFlow Aggregation**—A NetFlow feature that lets you summarize NetFlow export data on an IOS router before the data is exported to a NetFlow data collection system such as the NetFlow Collection Engine. This feature lowers bandwidth requirements for NetFlow export data and reduces platform requirements for NetFlow data collection devices.

**NetFlow Collection Engine** (formerly NetFlow FlowCollector)—Cisco application that is used with NetFlow on Cisco routers and Catalyst 5000 series switches. The NetFlow Collection Engine collects packets from the router that is running NetFlow and decodes, aggregates, and stores them. You can generate reports on various aggregations that can be set up on the NetFlow Collection Engine.

**NetFlow sampler**—NetFlow sampler map that has been applied to at least one physical interface or subinterface.

**NetFlow sampler map**—Defines a set of properties (such as the sampling rate) for NetFlow sampling.

**NetFlow v9**—NetFlow export format version 9. A flexible and extensible means to carry NetFlow records from a network node to a collector. NetFlow version 9 has definable record types and is self-describing for easier NetFlow Collection Engine configuration.

**options data record**—Special type of data record (which is based on an options template) with a reserved template ID that provides information about the NetFlow process itself.

**options template**—Type of template record used to communicate the format of data related to the NetFlow process.

**packet header**—First part of an export packet. It provides basic information about the packet (such as the NetFlow version, number of records contained in the packet, and sequence numbering) so that lost packets can be detected.

**template flowset**—Collection of one or more template records that are grouped in an export packet.

**template ID**—Unique number that distinguishes a template record from other template records produced by the same export device. A NetFlow Collection Engine application that receives export packets from several devices should be aware that uniqueness is not guaranteed across export devices. Thus, the NetFlow Collection Engine should also cache the address of the export device that produced the template ID in order to enforce uniqueness.

**template record**—Defines the format of subsequent data records that might be received in current or future export packets. A template record within an export packet does not necessarily indicate the format of data records within that same packet. A NetFlow Collection Engine application must cache any template records received and then parse any data records it encounters by locating the appropriate template record in the cache.

**ToS**—type of service byte. Second byte in the IP header that indicates the desired quality of service for a specific datagram.

**Note**

---

Refer to [Internetworking Terms and Acronyms](#) for terms not included in this glossary.

---

---

CCVP, the Cisco logo, and Welcome to the Human Network are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn is a service mark of Cisco Systems, Inc.; and Access Registrar, Aironet, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, LightStream, Linksys, MeetingPlace, MGX, Networkers, Networking Academy, Network Registrar, PIX, ProConnect, ScriptShare, SMARTnet, StackWise, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0711R)

---

© 2003–2005 Cisco Systems, Inc. All rights reserved.