CHAPTER 6

IP Accounting

This chapter describes the IP Accounting features in Cisco IOS and enables you to distinguish the different IP Accounting functions and understand SNMP MIB details. This chapter also provides a command-line reference.

IP Accounting is a very useful accounting feature in Cisco IOS, but it's not as well known as other features, such as NetFlow. The fact that Cisco has considered replacing IP Accounting by adding new features to NetFlow potentially turns IP Accounting into a corner case solution. However, compared to NetFlow, IP Accounting offers some advantages that make it an interesting feature to investigate: easy results retrieval via a MIB and limited resource consumption. Furthermore, access-list accounting currently cannot be solved with the NetFlow implementation. Note that NetFlow recently added the export of the MAC address as a new information element. Refer to coverage of NetFlow Layer 2 and the Security Monitoring Exports feature in Chapter 7, "NetFlow."

IP Accounting comes in four variations:

- Basic IP Accounting, which this book calls "IP Accounting (Layer 3)"
- IP Accounting Access Control List (ACL)
- IP Accounting MAC Address
- IP Accounting Precedence

Note that Cisco documentation is not always consistent for the different IP Accounting features. Therefore, this book uses the command-line interface (CLI) commands as titles, except for "IP Accounting Access Control List," where the related CLI command is **ip accounting access-violations**.

This chapter discusses in detail each flavor of IP Accounting, using a basic structure. First, the fundamentals are explained, followed by an overview of CLI operations, and then SNMP operations. It concludes by comparing the IP Accounting features to the questions raised in Chapter 2, "Data Collection Methodology":

- What to collect?
- Where and how to collect?
- How to configure?

- Who is the user?
- Potential scenarios.

IP Accounting (Layer 3)

IP Accounting (Layer 3) collects the number of bytes and packets processed by the network element on a source and destination IP address basis. Only transit traffic that enters and leaves the router is measured, and only on an outbound basis. Traffic generated by the router or traffic terminating in the router is not included in the accounting statistics. IP Accounting (Layer 3) collects individual IP address details, so it can be used to identify specific users for usage-based billing. To provide the operator with the opportunity of "snapshot" collections in the network, IP Accounting (Layer 3) maintains two accounting databases: an active database and a checkpoint database. The active collection process always updates the active database and therefore constantly increments the counters while packets pass the router. To get a snapshot of the traffic statistics, a CLI command or SNMP request can be executed to copy the current status from the active database to the checkpoint database. This copy request can be automated across the network to be executed at the same time, and a Network Management application can later retrieve the accounting details from the checkpoint database to present consistent accounting data to the operator. The checkpoint database offers a "frozen" snapshot of the complete network. Trying to achieve the same result by synchronously polling entire MIB tables across multiple network elements would introduce some inaccuracies, and hence no real "frozen" snapshots. The collected data can be used for performance and trending applications that require collections at regular intervals. The snapshot function is unique to IP Accounting.

IP Accounting (Layer 3) Principles

The principles of IP Accounting (Layer 3) can be summarized as follows:

- IP Layer 3 outbound (egress) traffic is collected.
- Only transit traffic that enters and leaves the router is collected; traffic that is generated by the router or terminated in the router is not included.
- IP Accounting (Layer 3) also collects IPX traffic. In this case, IPX source and destination addresses are reported instead of IP addresses.
- Egress MPLS core traffic collection is a new feature.
- Active and checkpoint databases enable "snapshot" collections.

- Collection data is accessible via CLI and SNMP; however, the initial configuration must be done via CLI. To retrieve the collection results via SNMP, you need to enable SNMP first. When configuring SNMP, distinguish between read-only access and read-write access. For more details about SNMP configuration, see Chapter 4, "SNMP and MIBs."
- The MIB contains only 32-bit SNMP counters.

Supported Devices and IOS Versions

The following list defines the devices and Cisco IOS Software releases that support IP Accounting (Layer 3):

- IP Accounting (Layer 3) was introduced in IOS 10.0.
- It is supported on all routers, including Route Switch Module (RSM) and Multilayer Service Feature Card (MSFC), except for the Cisco 12000. Note that IP Accounting cannot account for MLS-switched traffic on the Catalyst 6500/7600, so it collects only a subset of traffic on these platforms.
- It is supported on all physical interfaces and logical subinterfaces.
- IP Accounting (Layer 3) runs on the top of all switching paths, except for autonomous switching, silicon switching engine (SSE) switching, and distributed switching (dCEF) on the interface. On the Cisco 7500 router, IP Accounting (Layer 3) causes packets to be switched on the Route Switch Processor (RSP) instead of the Versatile Interface Processor (VIP), which can cause additional performance degradation.

CLI Operations

Notable commands for configuring, verifying, and troubleshooting IP Accounting (Layer 3) are as follows:

- router(config-if)# **ip accounting output-packets**
  enables IP Accounting (Layer 3) for output traffic on the interface.
- router(config)# **ip accounting-list** [**ip address**] [**ip address mask**]
  defines filters to control the hosts for which IP Accounting (Layer 3) information is kept. The filters are similar to an aggregation scheme and can be used to reduce the number of collected records. If filters are applied, details such as number of packets and bytes are kept only for the traffic that matches the filters, while all others are aggregated into "transit records."

- router(config)# **ip accounting-transits** *count*
  controls the number of transit records that are stored in the IP Accounting (Layer 3) database. Transit entries are those that do not match any of the filters specified by the global configuration command **ip accounting-list**. If no filters are defined, no transit entries are possible. The default number of transit records that are stored in the IP Accounting (Layer 3) database is 0.

  Note that the term "transit" in this case refers to packets that are not matched by the filter statements. In the IP Accounting (Layer 3) definition, "transit" refers to packets that traverse the router, compared to traffic that is generated at the router or destined for the router.

- router(config)# **ip accounting-threshold** *count*
  sets the maximum number of accounting entries to be created. The accounting threshold defines the maximum number of entries (source and destination address pairs) that are accumulated. The default accounting threshold is 512 entries, which results in a maximum table size of 12,928 bytes. The threshold counter applies to both the active and checkpoint tables.

  The threshold value depends on the traffic mix, because different traffic types create different records for the source and destination address pairs. Whenever the table is full, the new entries (overflows) are not accounted. However, **show ip accounting** displays the overflows: "Accounting threshold exceeded for X packets and Y bytes." Alternatively, these values are available in the MIB: actLostPkts (lost IP packets due to memory limitations) and actLostByts (total bytes of lost IP packets). You should monitor the overflows number, at least during the deployment phase, to find the right balance between the number of entries and memory consumption.

- router# **show ip accounting** [**checkpoint**] **output-packets**
  displays the active accounting or checkpoint database.

- router# **clear ip accounting**
  copies the content of the active database to the checkpoint database and clears the active database afterward.

- router# **clear ip accounting checkpoint**
  clears the checkpoint database.

NOTE The IP Accounting (Layer 3) and IP Accounting Access Control List entries share the same databases. Consequently, there is no explicit command to erase the IP Accounting (Layer 3) entries independently of the IP Accounting ACL entries.

SNMP Operations

The OLD-CISCO-IP-MIB has two tables:

- lipAccountingTable, the active database
- lipCkAccountingTable, the checkpoint database

The MIB variable actCheckPoint must be read first and then set to the same value that was read to copy the active database into the checkpoint database. After a successful SNMP set request, actCheckPoint is incremented by 1. Setting actCheckPoint is the equivalent of the **clear ip accounting** CLI command. A Network Management application can retrieve the MIB variable lipCkAccountingTable to analyze stable data in the checkpoint database. There is no SNMP variable to erase the content of the checkpoint database; however, setting actCheckPoint again flushes the checkpoint database and copies the content of the active database.

Details of the IP Accounting MIB (OLD-CISCO-IP-MIB) are as follows:

- **Active database**—The lipAccountingTable table contains four relevant components:
  - actSrc is the active database source.
  - actDst is the active database destination.
  - actPkts is the active database packets.
  - actByts is the active database bytes.

  The table indexes are actSrc and actDst.
- **Checkpoint database**—The lipCkAccountingTable table contains four relevant components:
  - ckactSrc is the checkpoint database source.
  - ckactdDst is the checkpoint database destination.
  - ckactPkts is the checkpoint database packets.
  - ckactByts is the checkpoint database bytes.

  The table indexes are ckactSrc and ckactDst.
- actCheckPoint MIB variable

NOTE The active and checkpoint MIB tables contain an ACL violations entry. Because it is relevant only to the IP Accounting Access Control List, it is not discussed in this section.

Examples (CLI and SNMP)

The following example provides a systematic introduction for configuring and monitoring IP Accounting (Layer 3) and displays the results for both CLI and SNMP.

Initial Configuration

Initially, both the active database (lipAccountingTable) and checkpoint database (lipCkAccountingTable) are empty, as shown from the router CLI and from the SNMP tables.

```
router#show ip accounting output-packets
   Source         Destination            Packets           Bytes
   Accounting data age is 0
router#show ip accounting checkpoint output-packet
   Source         Destination            Packets           Bytes
   Accounting data age is 0
```

The router is accessed with SNMP2c (SNMP version 2c), the read community string is public, and the SNMP tool net-snmp is used.

```
SERVER % snmpwalk -c public -v 2c <router> lipAccountingTable
    actDst.0.0.0.0.0.0.0.0 = IpAddress: 0.0.0.0
    actByts.0.0.0.0.0.0.0.0 = INTEGER: 0
SERVER % snmpwalk -c public -v 2c <router> lipCkAccountingTable
    ckactDst.0.0.0.0.0.0.0.0 = IpAddress: 0.0.0.0
    ckactByts.0.0.0.0.0.0.0.0 = INTEGER: 0
```

The IP Accounting (Layer 3) configuration is straightforward:

```
router(config)#int serial 0/0
router(config-if)#ip accounting output-packets
router(config-if)#exit
```

Collection Monitoring

After configuring IP Accounting (Layer 3), the active database populates:

```
router#show ip accounting output-packet
   Source         Destination            Packets         Bytes
 192.1.1.110      192.1.1.97                5             500
 192.1.1.110      192.1.1.26                5             500
```

The corresponding MIB table shows the identical entries:

```
SERVER % snmptable -Ci -Cb -c public -v 2c <router> lipAccountingTable
                     index         Src         Dst  Pkts  Byts
          192.1.1.110.192.1.1.26 192.1.1.110 192.1.1.26    5   500
          192.1.1.110.192.1.1.97 192.1.1.110 192.1.1.97    5   500
```

At this point, the checkpoint database is still empty. The active database content is cleared by copying its content to the checkpoint database:

```
router#clear ip accounting
```

As an alternative, the **clear ip accounting** mechanism can be mimicked by using the actCheckPoint MIB variable procedure. That means reading the content of the MIB variable and setting it again to the same value that was read:

```
SERVER % snmpget -c public -v 2c <router> actCheckPoint.0
       actCheckPoint.0 = INTEGER: 0
SERVER % snmpset -c private -v 2c <router> actCheckPoint.0 i 0
       actCheckPoint.0 = INTEGER: 0
      SERVER % snmpget -c public -v 2c <router> actCheckPoint.0
       actCheckPoint.0 = INTEGER: 1
```

The two entries just discussed are now in the checkpoint database, but the active database is empty:

```
router#show ip accounting output-packets
   Source         Destination           Packets            Bytes
   Accounting data age is 0
router#show ip accounting output-packets checkpoint output-packets
   Source         Destination           Packets            Bytes
 192.1.1.110      192.1.1.97               5                500
 192.1.1.110      192.1.1.26               5                500
SERVER % snmptable -Ci -Cb -c public -v 2c <router> lipCkAccountingTable
                         index         Src         Dst  Pkts  Byts
          192.1.1.110.192.1.1.26 192.1.1.110 192.1.1.26    5   500
          192.1.1.110.192.1.1.97 192.1.1.110 192.1.1.97    5   500
```

IP Accounting Access Control List (ACL)

The IP Accounting ACL identifies IP traffic that *fails* an IP access control list. This is a relevant security feature, because a sudden increase in traffic being blocked by an ACL can indicate a security attack in the network. Identifying IP source addresses that violate IP access control lists can help track down attackers. Alternatively, this mechanism can be used to identify when an attack is over, because an ACL is usually applied to block the attack. The data might also indicate that you should verify the network element configurations of the IP access control list. It is important to understand that the IP Accounting ACL does not account the amount of traffic that passes an individual ACL; therefore, it cannot be used for ACL optimization. However, the IP Accounting ACL can be used in conjunction with IP Accounting (Layer 3). For example, if ACLs are configured at a router, packets passing all ACLs are accounted by the IP Accounting (Layer 3) feature, and blocked traffic is collected via the IP Accounting ACL.

IP Accounting ACL is supported on ingress and egress traffic. There is no explicit command to recognize if the *incoming* traffic was blocked (the **ip access-group** *ACL-number* **IN** command, called ACL IN for simplicity) or the *outgoing* traffic was blocked (the **ip access-group** *ACL-number* **OUT** command, called ACL OUT). Nevertheless, because the blocking access list number is reported by the IP Accounting ACL, the direction can be identified in most cases.

NOTE There are rare exceptions where the network element modifies ACL-related parts of a packet and these packet changes relate to ACLs. An example is if an ACL blocks certain precedence values and Policy Routing or Committed Access Rate (CAR) is active (where the precedence could be changed). In a case where the ACL IN statement passes the original packet, but the ACL OUT statement blocks the modified packet, you cannot measure if ACL IN or ACL OUT has blocked the packet.

IP Accounting ACL Principles

The principles of IP Accounting ACL can be summarized as follows:

- It provides information for identifying IP traffic that fails IP ACLs.
- It supports standard and extended ACLs, but not named ACLS.
- It supports ACLs applied as both ingress and egress.
- It has the same database concept as IP Accounting (Layer 3): active and checkpoint databases.
- If the packet is blocked by an ACL, only the IP Accounting ACL database is updated, not the IP Accounting (Layer 3) database.
- Collection data is accessible via CLI and SNMP; however, the initial configuration is required via CLI. To retrieve the collection results via SNMP, you need to enable SNMP on the router first. When configuring SNMP, distinguish between read-only access and read-write access. For more details about SNMP configuration, see Chapter 4.
- The MIB contains only 32-bit SNMP counters.

Supported Devices and IOS Versions

The following list defines the devices and Cisco IOS Software releases that support IP Accounting ACL:

- IP Accounting ACL was introduced in IOS 10.3.
- It is supported on all routers, including the RSM and MSFC, except for the Cisco 12000.
- It is supported on all physical interfaces and logical subinterfaces.
- It is supported on all switching paths, except for autonomous switching, SSE switching, and dCEF. On the Cisco 7500 router, the IP Accounting ACL causes packets to be switched on the RSP instead of the VIP, which can cause performance degradation.

CLI Operations

Notable commands for configuring, verifying, and troubleshooting IP Accounting ACL are as follows:

- router(config-if)# **ip accounting access-violations**
  allows IP Accounting ACL to identify IP traffic that fails IP ACLs.
- router(config)# **ip accounting-threshold** *count*
  sets the maximum number of accounting entries to be created. The accounting threshold defines the maximum number of entries (source and destination address pairs) that are accumulated, with a default of 512 entries.

- router# **show ip accounting** [**checkpoint**] **access-violations**
  displays the active accounting or checkpoint database for ACL violations.
- router# **clear ip accounting**
  copies the content of the active database to the checkpoint database and afterwards clears the active database for both IP Accounting (Layer 3) and IP Accounting ACL.
- router# **clear ip accounting checkpoint**
  clears the checkpoint database.

NOTE The IP Accounting ACL and IP Accounting (Layer 3) entries share the same databases. Consequently, there is no explicit command to erase the IP Accounting ACL entries independently of the IP Accounting (Layer 3) entries.

SNMP Operations

IP Accounting ACL uses the same MIB (OLD-CISCO-IP-MIB) as IP Accounting (Layer 3), which was described previously. IP Accounting ACL cannot be configured via SNMP, but a copy function from the active database to the checkpoint database is provided. First, you should copy the active database to the checkpoint database. Afterward, retrieve the data from the checkpoint database if you want to collect consistent accounting data across multiple network elements at the same time. IP Accounting ACL uses the same MIB tables as IP Accounting (Layer 3) but augments the information with the ACL Identifier—the ACL number that was violated by packets from source to destination. The ACL identifier is represented by actViolation in the lipAccountingTable table (the active database) and ckactViolation in the lipCkAccountingTable table (the checkpoint database). actCheckPoint copies the active database into the checkpoint database. For details about the MIB configuration, see the section "SNMP Operations" under "IP Accounting (Layer 3)."

Examples (CLI and SNMP)

The following example for IP Accounting ACL provides CLI and SNMP commands and extends them to display the entries of the IP Accounting (Layer 3) database as well. This helps you understand the close relationship between IP Accounting (Layer 3) and IP Accounting ACL.

Initial Configuration

Initially, the active and checkpoint databases are empty for both IP Accounting (Layer 3) and IP Accounting ACL. An SNMP query to the lipAccountingTable MIB table (active) and to the lipAcCkAccountingTable MIB table (checkpoint) confirms this:

```
router#show ip accounting output-packets
   Source        Destination       Packets        Bytes
   Accounting data age is 0
```

```
router#show ip accounting access-violations
   Source        Destination       Packets        Bytes      ACL
   Accounting data age is 0

router#show ip accounting checkpoint output-packets
   Source        Destination       Packets        Bytes
   Accounting data age is 0

router#show ip accounting checkpoint access-violations
   Source        Destination       Packets        Bytes      ACL
   Accounting data age is 0
```

For this example, IP Accounting ACL is configured in addition to IP Accounting (Layer 3); however, it can be configured independently of IP Accounting (Layer 3). An access list is inserted, which blocks the traffic coming from the source IP address 192.1.1.110 and going to the destination IP address 192.1.1.97:

```
router(config)#access-list 107 deny ip host 192.1.1.110 host 192.1.1.97
router(config)#access-list 107 permit ip any any
router(config)#int serial 0/0
router(config-if)#ip accounting output-packets
router(config-if)#ip accounting access-violations
router(config-if)#ip access-group 107 out
router(config-if)#exit
```

Collection Monitoring

Afterwards, the following results can be retrieved from the router:

```
router#show ip accounting access-violations
   Source        Destination         Packets         Bytes   ACL
 192.1.1.110     192.1.1.97             3             300    107
  Accounting data age is 3
router#show ip accounting output-packets
   Source        Destination         Packets         Bytes
 192.1.1.110     192.1.1.26             5             500
  Accounting data age is 3
```

Three packets from the traffic (192.1.1.110, 192.1.1.97) are blocked by access list 107 and therefore are accounted by the IP Accounting ACL. The traffic (192.1.1.110, 192.1.1.26) is accounted by IP Accounting (Layer 3).

For the SNMP example, the router is accessed with SNMP2c, a read community string public, and the net-snmp SNMP utility. The entries from both IP Accounting (Layer 3) and the IP Accounting ACL appear in the same MIB lipAccountingTable table. The only difference is that the entries from the IP Accounting ACL will have a non-null value in the actViolation MIB variable:

```
SERVER % snmpwalk -c public -v 2c <router> lipAccountingTable
actSrc.192.1.1.110.192.1.1.26 = IpAddress: 192.1.1.110
actSrc.192.1.1.110.192.1.1.97 = IpAddress: 192.1.1.110
actDst.192.1.1.110.192.1.1.26 = IpAddress: 192.1.1.26
actDst.192.1.1.110.192.1.1.97 = IpAddress: 192.1.1.97
actPkts.192.1.1.110.192.1.1.26 = INTEGER: 5
actPkts.192.1.1.110.192.1.1.97 = INTEGER: 3
actByts.192.1.1.110.192.1.1.26 = INTEGER: 500
```

```
actByts.192.1.1.110.192.1.1.97 = INTEGER: 300
actViolation.192.1.1.110.192.1.1.26 = INTEGER: 0
actViolation.192.1.1.110.192.1.1.97 = INTEGER: 107
```

Formatting the result in a different way, the distinction between **show ip accounting output-packets** and **show ip accounting access-violations** is clear.

The first entry (**show ip accounting output-packets**) is

```
actSrc.192.1.1.110.192.1.1.26 = IpAddress: 192.1.1.110
actDst.192.1.1.110.192.1.1.26 = IpAddress: 192.1.1.26
actPkts.192.1.1.110.192.1.1.26 = INTEGER: 5
actByts.192.1.1.110.192.1.1.26 = INTEGER: 500
actViolation.192.1.1.110.192.1.1.26 = INTEGER: 0
```

The second entry (**show ip accounting access-violations**) is

```
actSrc.192.1.1.110.192.1.1.97 = IpAddress: 192.1.1.110
actDst.192.1.1.110.192.1.1.97 = IpAddress: 192.1.1.97
actPkts.192.1.1.110.192.1.1.97 = INTEGER: 3
actByts.192.1.1.110.192.1.1.97 = INTEGER: 300
actViolation.192.1.1.110.192.1.1.97 = INTEGER: 107
```

NOTE In this example, the first entry corresponds to **show ip accounting output-packets** and the second to **show ip accounting access-violations**. This correspondence is a coincidence in this case.

At this point, the checkpoint database is still empty. The **clear ip accounting** CLI command or the actCheckPoint MIB variable procedure copies the active database content to the checkpoint database. The two entries just discussed are now in the checkpoint database, and the active database is empty.

```
router#show ip accounting checkpoint access-violations
   Source        Destination           Packets            Bytes   ACL
 192.1.1.110     192.1.1.97              3                300    107
  Accounting data age is 10

router#show ip accounting checkpoint output-packets
   Source        Destination           Packets            Bytes
 192.1.1.110     192.1.1.26              5                500
  Accounting data age is 10

router#show ip accounting access-violations
   Source        Destination           Packets            Bytes   ACL
  Accounting data age is 11

router#show ip accounting output-packets
   Source        Destination           Packets            Bytes
  Accounting data age is 11

SERVER % snmpwalk -c public -v 2c <router> lipcKAccountingTable
ckactSrc.192.1.1.110.192.1.1.26 = IpAddress: 192.1.1.110
ckactSrc.192.1.1.110.192.1.1.97 = IpAddress: 192.1.1.110
ckactDst.192.1.1.110.192.1.1.26 = IpAddress: 192.1.1.26
ckactDst.192.1.1.110.192.1.1.97 = IpAddress: 192.1.1.97
```

```
ckactPkts.192.1.1.110.192.1.1.26 = INTEGER: 5
ckactPkts.192.1.1.110.192.1.1.97 = INTEGER: 3
ckactByts.192.1.1.110.192.1.1.26 = INTEGER: 500
ckactByts.192.1.1.110.192.1.1.97 = INTEGER: 300
ckactViolation.192.1.1.110.192.1.1.26 = INTEGER: 0
ckactViolation.192.1.1.110.192.1.1.97 = INTEGER: 107
```

IP Accounting MAC Address

IP Accounting MAC Address is comparable to the IP Accounting (Layer 3) feature. However, MAC addresses are collected instead of IP addresses, and there is no concept of a checkpoint database. IP Accounting MAC Address calculates the total number of packets and bytes for IP traffic on LAN interfaces, based on the source and destination MAC addresses. It also records a time stamp for the last packet received or sent. This feature helps the operator determine how much traffic is exchanged with various peers at Layer 2 exchange points, such as an Internet peering point. IP Accounting MAC Address collects individual MAC addresses, so it can be used to identify a specific user for usage-based billing. It also helps security administrators identify a sender's MAC address in case of an attack with faked IP addresses.

The maximum number of MAC addresses that can be stored at the network element for each physical interface is 512 entries for input and an additional 512 MAC addresses for output traffic. After the maximum is reached, subsequent MAC addresses are ignored. To keep addresses from not being taken into account, you should constantly check the number of available entries in the network element's local database and clear entries if it's getting close to 512.

IP Accounting MAC Address Principles

The principles of IP Accounting MAC Address can be summarized as follows:

- Inbound and outbound traffic statistics are collected per MAC address.
- Only LAN interfaces and subinterfaces (Ethernet, FastEthernet, FDDI, and VLAN) are supported.
- A time stamp is recorded (or updated) when the last packet is sent or received.
- When IP Accounting MAC Address is enabled, header compression is turned off so that the MAC information can be extracted from the header. When IP Accounting MAC Address is turned off, header compression is enabled.
- There is no concept of a checkpoint database.
- The maximum number of entries per physical interface and per direction (incoming or outgoing) is 512.

- Collection data is accessible via CLI and SNMP. However, all configuration changes must be done via CLI, because the CISCO-IP-STAT-MIB has no read-write parameters. To retrieve the collection results via SNMP, you need to enable SNMP on the network element first. For more details about SNMP configuration, see Chapter 4.
- The MIB contains 32-bit and 64-bit SNMP counters.

Supported Devices and IOS Versions

The following devices and Cisco IOS Software releases support IP Accounting MAC Address:

- IP Accounting MAC Address was introduced in IOS 11.1CC.
- It is supported on Ethernet, FastEthernet, FDDI, and VLAN interfaces. It works in conjunction with Cisco Express Forwarding (CEF), distributed Cisco Express Forwarding (dCEF), flow, and optimum switching.
- It is supported on all routers, including the MSFC, but not the RSM.
- On the Cisco 12000 router, it is supported only by the 3-port Gigabit Ethernet line cards.

CLI Operations

Notable commands for configuring, verifying, and troubleshooting IP Accounting MAC Address are as follows:

- router(config-if)# **ip accounting mac-address** {**input** | **output**}, where:
  - **input** performs accounting based on the source MAC address on received packets.
  - **output** performs accounting based on the destination MAC address on transmitted packets.
- router# **show interface** [*type number*] **mac-accounting**
  displays information for all interfaces configured for MAC accounting. To display information for a single interface, use the appropriate information for the *type number* arguments.
- router# **clear counters** [*interface-type interface-number*]
  clears all the interface counters. Because the IP Accounting MAC Address entries are stored per interface, the **clear counters** command clears the number of bytes and packets for each IP Accounting MAC Address entry in the output of **show interface** [*type number*] **mac-accounting**. However, the **clear counters** command does not remove any IP Accounting MAC Address entries. In the output from **show interface** [*type number*] **mac-accounting**, **clear counters** keeps the value of the time stamp for the last packet sent or received for that entry. The **clear counters** command does not clear the MIB counters, because SNMP counters can never be cleared, and it does not remove any IP Accounting MAC Address entries in the MIB table. An analogy is the **clear counters** command that clears the number of bytes and packets in the output of

show interface while the SNMP counters in the ifTable are not cleared. Note also that the **clear counters** command is applicable globally for all interfaces or for a single interface.

SNMP Operations

IP Accounting MAC Address uses the Cisco IP Statistics MIB to collect incoming and outgoing packets and bytes per MAC address. There is a maximum of 512 entries per physical interface per direction (ingress or egress)**.** You have to use the CLI to enable and disable IP Accounting MAC Address. Entries can be read but not deleted via SNMP. They can be deleted using the CLI command **clear counters** instead. The CISCO-IP-STAT-MIB (Cisco IP Statistics MIB) was updated to support 32-bit and 64-bit counters. For high-speed interfaces, 64-bit counters are relevant, because on a 1-Gigabit interface, a 32-bit counter wraps after 34 seconds.

The IP Accounting MAC Address part of the MIB consists of two tables with separate 32-bit counters and 64-bit counters, plus an extra table for the number of free entries in the database:

- **cipMacTable** is the MAC table for 32-bit counters, where an entry is created for each unique MAC address that sends or receives IP packets. It contains four variables:
  - **cipMacDirection** is the object's data source.
  - **cipMacAddress** is the MAC address.
  - **cipMacSwitchedPkts** is the counter in packets with respect to cipMacAddress.
  - **cipMacSwitchedBytes** is the counter in bytes with respect to cipMacAddress.

  The table indexes are ifIndex, cipMacDirection, and cipMacAddress.
- **cipMacXTable** is the extended MAC table for 64-bit counters, which contains only two entries.
  - **cipMacHCSwitchedPkts** is the high-capacity counter in packets with respect to cipMacAddress. This object is the 64-bit version of cipMacSwitchedPkts.
  - **cipMacHCSwitchedBytes** is the high-capacity counter in bytes with respect to cipMacAddress. This object is the 64-bit version of cipMacSwitchedBytes.

  The table indexes are ifIndex, cipMacDirection, and cipMacAddress.
- **cipMacFreeTable** specifies the number of available entries in the database.
- **cipMacFreeCount** is the number of items in the MAC free space.

The table indexes are ifIndex and cipMacFreeDirection.

Examples (CLI and SNMP)

The following example provides a systematic introduction to configuring and monitoring IP Accounting MAC Address and displays the results for both CLI and SNMP.

Initial Configuration

Initially, there are no IP Accounting MAC Address entries.

In this configuration, both IP Accounting MAC Address input and output are enabled:

```
router(config-if)#interface fastethernet 0/0
router(config-if)#ip accounting mac-address input
router(config-if)#ip accounting mac-address output
router(config-if)#exit
```

Collection Monitoring

The entries populate:

```
Router#show interface mac-accounting
FastEthernet1/0 Eth -> Nms-bb-1: Port 4/20
      Input (504 free)
0010.8305.c421(115): 7 packets, 590 bytes, last: 95924ms ago.
.
.
.
                    Total:  111 packets, 10290 bytes
      Output  (504 free)
0800.2087.66c1(8 ): 2 packets, 375 bytes, last: 8520ms ago
.
.
.
                    Total:  39 packets, 5536 bytes
```

For clarity, only the first input and output entries are displayed. The corresponding MIB table shows the identical entries, only one of which is displayed:

```
SERVER % snmpwalk -c public -v 2c martel cipMacTable
cipMacSwitchedPkts.9.input.0.16.131.5.196.33 : Counter: 7
cipMacSwitchedBytes.9.input.0.16.131.5.196.33 : Counter: 590
```

The table indexes are as follows:

- ifIndex is 9 in this case, which represents fastethernet 1/0:

  ```
  Router #show snmp mib ifmib ifIndex fastethernet 1/0
      Interface = fastethernet 1/0, ifIndex =9
  ```

- cipMacDirection is input or output.
- cipMacAddress, where 0.16.131.5.196.33 is the MAC address, such as 0010.8305.c421.

This SNMP entry corresponds to the following entry in the **show** command:

```
0010.8305.c421(115): 7 packets, 590 bytes, last: 95924ms ago.
```

NOTE The information about the last time a packet was observed from/to the specific MAC address is not available in the MIB—only from the **show** command.

The SNMP request confirms that 504 entries are available:

```
SERVER % snmpwalk -c public -v 2c <router> cipMacFreeTable
CISCO-IP-STAT-MIB::cipMacFreeCount.9.input = Gauge32: 504
CISCO-IP-STAT-MIB::cipMacFreeCount.9.output = Gauge32: 504
```

In a situation where the counters are small, polling cipMacXTable, which contains the high-capacity counter counter64, would return the same results as polling cipMacTable.

Finally, the IP MAC address counters can be cleared, either specifically for the interface or globally for all interfaces, but no entries are deleted:

```
Router(config)#clear counters [fastethernet 1/0]
Router#show interface mac-accounting
FastEthernet1/0 Eth -> Nms-bb-1: Port 4/20
      Input  (504 free)
0010.8305.c421(115): 0 packets, 0 bytes, last: 125876ms ago
```

In the preceding example, the counters for packets and bytes are reset to 0. All other entries, along with the content of the "last" field, are preserved. The **clear counters** CLI command has no effect on the MIB's content.

NOTE The **clear counters** command affects both the IP Accounting Precedence and IP Accounting MAC Address counters. This could be considered a limitation when enabled on the same interface.

IP Accounting Precedence

The IP Accounting Precedence feature provides IP precedence-related traffic accounting information. The collection per interface consists of the total number of packets and bytes for each of the eight IP Precedence values, separately per direction (send and receive).

IP Accounting Precedence does not collect individual IP or MAC addresses, so it cannot be used to identify a specific user for usage-based billing, except in cases where the (sub)interface can serve as user identifier. There is no concept of a checkpoint database. Regarding QoS operations, it is important to distinguish between ingress and egress traffic on an interface:

- For incoming packets on the interface, the accounting statistics are gathered before input CAR/Distributed CAR (DCAR) is performed on the packet. Therefore, if CAR/DCAR changes the precedence on the packet, it is counted based on the old precedence setting with the **show interface precedence** command.
- For outgoing packets on the interface, the accounting statistics are gathered after output features such as DCAR, Distributed Weighted Random Early Detection (DWRED), and Distributed Weighted Fair Queuing (DWFQ) are performed on the packet.

IP Accounting Precedence Principles

The principles of IP Accounting Precedence can be summarized as follows:

- Both inbound and outbound traffic is collected for eight IP precedence classes.
- IP Accounting Precedence is supported on physical interfaces and subinterfaces.
- There is no concept of a checkpoint database.
- Individual IP or MAC addresses are not collected.
- Collection data is accessible via CLI and SNMP. However, all configuration changes need to be done via CLI, because the CISCO-IP-STAT-MIB has no read-write parameters.
- The MIB contains 32-bit and 64-bit SNMP counters.
- To retrieve the collection results via SNMP, you have to enable SNMP on the network element first. When configuring SNMP, distinguish between read-only access and read-write access. For more details about SNMP configuration, see Chapter 4.

Supported Devices and IOS Versions

The following list defines the devices and Cisco IOS Software releases that support IP Accounting Precedence:

- IP Accounting Precedence was introduced in IOS 11.1CC.
- It is supported on CEF, dCEF, and optimum switching.
- It supports Virtual Routing and Forwarding (VRF) interfaces but does not support tunnel interfaces.
- It is supported on all routers, including the RSM and MSFC, except for the Cisco 12000.

CLI Operations

Notable commands for configuring, verifying, and troubleshooting IP Accounting Precedence are as follows:

- router(config-if)# **ip accounting precedence** {**input** | **output**}, where:
  - **input** performs accounting based on IP precedence on received packets.
  - **output** performs accounting based on IP precedence on transmitted packets.
- router# **show interface** [*type number*] **precedence**
  displays information for all interfaces configured for IP Accounting Precedence. To display information for a single interface, enter the appropriate values for the *type number* arguments.
- router# **clear counters** [*interface-type interface-number*]
  clears interface counters. Because the IP Accounting Precedence entries are stored per interface, the **clear counters** command clears all IP Accounting Precedence entries. Note that this function is different from the IP Accounting MAC Address feature. The **clear counters** command does not delete the content of the cipPrecedenceTable MIB table. An analogy would be the **clear counters** command that clears the number of bytes and packets in the output of **show interface** while the SNMP counters in the ifTable are not cleared. Note also that the **clear counters** command is applicable globally for all interfaces or for a single interface.

SNMP Operations

IP Accounting Precedence can be configured only by using the CLI. Collection data can be read but not deleted via SNMP. It can be deleted using the CLI command **clear counters**. CISCO-IP-STAT-MIB supports 32-bit and 64-bit counters.

The IP Accounting Precedence part of the MIB consists of two tables with separate 32-bit counters and 64-bit counters:

- **cipPrecedenceTable**—The 32-bit counter table for IP Accounting Precedence contains four variables:
  - **cipPrecedenceDirection** is the object's data source (incoming or outgoing traffic).
  - **cipPrecedenceIpPrecedence** is the IP precedence value this object is collected on, with a total of eight different precedence values (0 to 7).
  - **cipPrecedenceSwitchedPkts** is the number of packets per IP precedence value (cipPrecedenceIpPrecedence).
  - **cipPrecedenceSwitchedBytes** is the number of bytes per IP precedence value (cipPrecedenceIpPrecedence).

  The indexes of the three tables are ifIndex, cipPrecedenceDirection, and cipPrecedenceIpPrecedence

- **cipPrecedenceXTable**—The 64-bit counter extension table for IP Accounting Precedence contains two variables:
  - **cipPrecedenceHCSwitchedPkts** is the number of packets per IP precedence value (cipPrecedenceIpPrecedence). This object is the 64-bit version of cipPrecedenceSwitchedPkts.
  - **cipPrecedenceHCSwitchedBytes** is the number of bytes per IP precedence value (cipPrecedenceIpPrecedence). This object is the 64-bit version of cipPrecedenceSwitchedBytes.

  The three table indexes are ifIndex, cipPrecedenceDirection, and cipPrecedenceIpPrecedence.

Examples (CLI and SNMP)

The following example provides a systematic introduction to configuring and monitoring IP Accounting Precedence and displays the results for both CLI and SNMP.

Initial Configuration

Initially, there are no IP Accounting Precedence entries.

In this configuration, both IP Accounting Precedence input and output are enabled:

```
router(config-if)#interface serial 0/0
router(config-if)#ip accounting precedence input
router(config-if)#ip accounting precedence output
router(config-if)#exit
```

Collection Monitoring

The entries populate:

```
router#show interfaces precedence
Serial0/0
  Input
           Precedence 6:  8 packets, 467 bytes
  Output
           Precedence 0:  6 packets, 504 bytes
           Precedence 6:  11 packets, 863 bytes
```

The corresponding MIB table shows the identical entries.

The router is accessed with SNMP2c (SNMP version 2c), the read community string is public, and the SNMP tool **net-snmp** is used:

```
SERVER % snmpwalk -c public -v 2c <router> cipPrecedenceTable
cipPrecedenceSwitchedPkts.1.input.0 = Counter32: 0
cipPrecedenceSwitchedPkts.1.input.1 = Counter32: 0
cipPrecedenceSwitchedPkts.1.input.2 = Counter32: 0
```

```
cipPrecedenceSwitchedPkts.1.input.3 = Counter32: 0
cipPrecedenceSwitchedPkts.1.input.4 = Counter32: 0
cipPrecedenceSwitchedPkts.1.input.5 = Counter32: 0
cipPrecedenceSwitchedPkts.1.input.6 = Counter32: 8
cipPrecedenceSwitchedPkts.1.input.7 = Counter32: 0
cipPrecedenceSwitchedPkts.1.output.0 = Counter32: 6
cipPrecedenceSwitchedPkts.1.output.1 = Counter32: 0
cipPrecedenceSwitchedPkts.1.output.2 = Counter32: 0
cipPrecedenceSwitchedPkts.1.output.3 = Counter32: 0
cipPrecedenceSwitchedPkts.1.output.4 = Counter32: 0
cipPrecedenceSwitchedPkts.1.output.5 = Counter32: 0
cipPrecedenceSwitchedPkts.1.output.6 = Counter32: 11
cipPrecedenceSwitchedPkts.1.output.7 = Counter32: 0
cipPrecedenceSwitchedBytes.1.input.0 = Counter32: 0
cipPrecedenceSwitchedBytes.1.input.1 = Counter32: 0
cipPrecedenceSwitchedBytes.1.input.2 = Counter32: 0
cipPrecedenceSwitchedBytes.1.input.3 = Counter32: 0
cipPrecedenceSwitchedBytes.1.input.4 = Counter32: 0
cipPrecedenceSwitchedBytes.1.input.5 = Counter32: 0
cipPrecedenceSwitchedBytes.1.input.6 = Counter32: 467
cipPrecedenceSwitchedBytes.1.input.7 = Counter32: 0
cipPrecedenceSwitchedBytes.1.output.0 = Counter32: 504
cipPrecedenceSwitchedBytes.1.output.1 = Counter32: 0
cipPrecedenceSwitchedBytes.1.output.2 = Counter32: 0
cipPrecedenceSwitchedBytes.1.output.3 = Counter32: 0
cipPrecedenceSwitchedBytes.1.output.4 = Counter32: 0
cipPrecedenceSwitchedBytes.1.output.5 = Counter32: 0
cipPrecedenceSwitchedBytes.1.output.6 = Counter32: 863
cipPrecedenceSwitchedBytes.1.output.7 = Counter32: 0
```

The table indexes are as follows:

- ifIndex

  In this case it is 1, which represents serial 0/0:

  ```
  Router #show snmp mib ifmib ifIndex serial 0/0
            Interface = Serial0/0, ifIndex = 1
  ```

- cipPrecedenceDirection: input or output
- cipPrecedenceIpPrecedence: a value from 0 to 7

For example, the entry (Input, Precedence 6, 8 packets, 467 bytes) is represented in the SNMP table by

```
cipPrecedenceSwitchedBytes.1.input.6 = Counter32: 467
```

In a situation where the counters are small, polling cipPrecedenceXTable, which contains the high-capacity counter counter64, returns the same results as polling cipPrecedenceTable.

Finally, the IP Accounting Precedence counters can be cleared, either specifically for the interface or globally for all interfaces:

```
router(config)#clear counters [serial 0/0]
router#show interfaces precedence
Serial0/0
  Input
           none
        Output
           none
```

NOTE The **clear counters** affects the IP Accounting Precedence counters and the IP Accounting MAC Address counters. This could be considered a limitation when enabled on the same interface.

Applicability

Table 6-1 compares the four different variations of IP Accounting and relates them to the fundamental questions that were discussed in Chapter 2.

Table 6-1 *Comparison of the Different IP Accounting Features*

Criterion	IP Accounting (Layer 3)	IP Accounting ACL	IP Accounting MAC Address	IP Accounting Precedence
What to collect (data collection details)?	Total traffic (packets and bytes) per: IP source address IP destination address	Total traffic (packets and bytes) per failed ACL: IP source address IP destination address ACL identifier	Total traffic (packets and bytes) per: Source MAC address Destination MAC address	Total traffic (packets and bytes) per: Eight different IP precedence values
Where to collect?	Edge or core, LAN or WAN (any interface type)	Edge or core, LAN or WAN (any interface type)	Edge: LAN interface types only	Edge or core, LAN or WAN (any interface type)
How to configure?	CLI (scripts)	CLI (scripts)	CLI (scripts)	CLI (scripts)
How to collect?	Pull model: CLI (scripts) SNMP	Pull model: CLI (scripts) SNMP	Pull model: CLI (scripts) SNMP	Pull model: CLI (scripts) SNMP
How to identify the user?	IP address	IP address	MAC address	Not available
Use case example	Network monitoring and planning, user monitoring, usage-based billing	Security analysis	Security analysis, network monitoring and planning, user monitoring, usage-based billing, peering agreements	Network QoS monitoring and planning
Additional details	Only 32-bit SNMP counters Two different databases provide a global "snapshot" of the network	Only 32-bit SNMP counters Two different databases provide a global "snapshot" of the network	32-bit and 64-bit SNMP counters	32-bit and 64-bit SNMP counters