CHAPTER 4

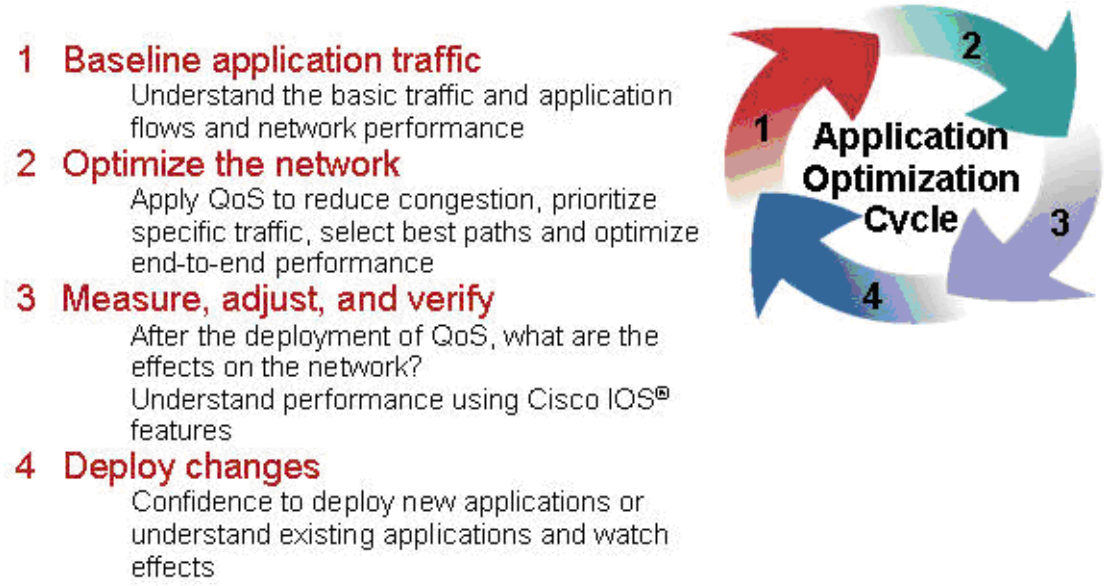
Cisco Monitoring Instrumentation

Understanding and addressing application performance issues brings visibility into how the business actually uses the network resources, and with abilities to measure how well applications are performing.

This chapter summarizes the key monitoring instrumentation technologies that provide essential information and sources of data for meeting the needs of the key performance management disciplines that optimize the networks and applications. Chapter 8 will describe the performance monitoring tools that consume this monitoring instrumentation data.

Figure 4-1 outlines a general process that can be used to incrementally increase understanding of one's network and progressively deploy measurable improvements and adjustments as required.

Figure 4-1 WAN and Application Optimization Life Cycle



4.1 Profiling and Baselining

The first step to WAN and application optimization is to profile network activity by establishing a reference from which service quality and application delivery effectiveness can be measured.

The profile of a network describes the traffic patterns and resource bottlenecks of a network. This identifies for the network operator the links and protocols that are the best candidates for optimization. Through profiling, a network engineer can focus on only those network components whose optimization will help improve and develop baselines as a performance benchmark.

Baselining is the establishment of acceptable network behavior. This includes understanding available bandwidth, identifying a normal pattern of network behavior such as network delays and what applications are running on the network, understanding each application's behavior (and requirements) on the network, and measuring application response times. For example, while not consistent with a daily average, baselining should capture and account for behaviors such as non-working weekend days that are less stressful on the network. Network administrators need to know the acceptable range for network performance before they can make reliable conclusions about possible performance degradation. With proper baselining, administrators can differentiate between consistent network behavior and anomalous (candidates for improvement) network behavior.

A few of the goals in baselining follow:

4.1.1 Ensure Network Stability

Complete internetwork communications can be easily obstructed if a network device such as a server or a single segment in a LAN becomes unreachable. The same is true if a server behind a router within the campus LAN environment or even behind the WAN cannot be contacted. Many different scenarios can cause problems in a large network and being able to maintain stability is a paramount concern of network managers.

4.1.2 Ensure Network Reliability

Many upper-layer applications present in today's enterprise networks require connection-based processing during communications from one device to another. Maintaining a consistent connection is essential when critical communications take place between network devices, such as a workstation and a server. Being able to maintain low latency between a database and client machine, for instance, would be very important for applications that rely on constant access to the database.

Cisco IOS instrumentation provides a good starting point for creating a network performance baseline through the following components:

- NetFlow
- IPSLA
- NBAR
- CBQoS MIB

NetFlow provides a good source of traffic flow information for capturing normal and abnormal behaviors on the network. Additionally, standardized SNMP MIBs from individual devices provide basic information about the network such as traffic volume by byte, errors, utilization on interfaces, etc. NBAR, a traffic identification and classification engine built into IOS, can discover the types of applications that are present on the network. Together, NetFlow, MIBs, and NBAR provide a comprehensive baseline about the physical network and the paths application flows take as they utilize the network.

Creating response time baseline is important to the success of an IT organization in establishing service quality levels. Active and passive response time measurements are two methodologies for measuring application response times. Cisco IP SLA is the active method. Cisco WAAS Flow agent, Cisco NAM and NetQoS SuperAgent implement the passive method.

There is no one single source of information for baselining your network and applications. IT organizations will need to use different monitoring instrumentation data in order to gain a solid understanding of the normal behavior of the applications, the network, and IT resources.

4.1.3 Optimize the Network

Once you have end-to-end visibility of the network and the applications, you can then determine which optimization tools and technologies to utilize to best meet the requirements. The second step is to apply the optimization or control techniques to enhance application performance.

4.1.4 Measure, Adjust, and Verify

The third step is to assess the effectiveness of each successive WAN optimization initiative. This includes continuously monitoring and collecting information about the network and application behavior, and comparing the behavior before and after successive WAN optimization initiatives.

For example, when new QoS policies have just been deployed, you want to measure the effects of the network. CBQoS MIB from individual devices provide information about the network before and after applying the QoS policies. Similarly, after deploying WAAS, you want to determine the effectiveness of WAAS before and after compression and acceleration. WAAS Flow agent provides such information.

Measuring application response times for key applications both before and after WAN optimization and control techniques allows IT organizations to determine if the changes achieve desirable results. At the same time, it allows IT organizations to determine if the changes cause unacceptable impact on the company's other key applications.

Together, CBQoS, MIB, WAAS Flow agent, IP SLA, and NAM can serve as useful tools for measurement, adjustment, and verification of WAN optimization initiatives.

4.1.5 Deploy Changes

The fourth step is to deploy changes. IT organizations regularly deploy new applications and updates to existing applications to meet changing business needs. As new applications are deployed or changes are made, new baselines need to be established. The application optimization cycle must start all over again.

4.2 Monitoring Instrumentation Overview

Continuous performance monitoring is key to optimized application performance. Whether traffic is generated synthetically and metrics from an end host generating and receiving traffic is monitored actively, or natural network traffic is monitored passively but with lower network overhead, network and application performance data can be retrieved from a wide variety of data sources, each offering a different level of granularity and relative value. The subsequent subsections provide detail description of key monitoring instrumentation.

As networks grow in size and complexity and enterprise requirements grow, a need for greater visibility arises. IT directors and managers need tools that can help identify the various segments of their network that need improvement to allow a more efficient distribution of limited budget resources. Cisco products come packaged with tools that provide the platform to build detailed network monitoring abilities.

4.3 IOS Instrumentation

This section describes monitoring information built into IOS, such as:

- Cisco IP service level agreements (IP SLA)
- Cisco NetFlow
- NBAR
- CBQoS MIB

4.3.1 IP SLA

IP SLA is a feature set in Cisco IOS software that enables users to analyze service levels for IP applications and services. IP SLA uses reliable, scheduled continuous traffic generation to measure network performance. IP SLAs can perform network assessments, verify quality of service (QoS), ease the deployment of new services, and assist administrators with network troubleshooting.

Important IP SLA highlights include:

- Monitoring network performance:
  - Ability to measure jitter, packet loss, packet ordering, packet corruption and delay
- Network availability monitoring:
  - Test connectivity of network resources
- Network troubleshooting:
  - Troubleshoot network elements through consistent and reliable measurement

IP SLA has two key components: a source device that generates, receives, and analyzes traffic, and the target device for which SLA measurements are gathered. Additional accuracy and detail for the measurements can be achieved using the optional IP SLA Responder function on the target device. The IP SLA responder enables the target device to mark the arrival and departure times of SLA probes, so that any local processing latency on the responder is mitigated. For example, with regular ICMP echo and echo reply, the echo target can choose to process ICMP traffic in a slow, deprioritized path. Without the SLA responder-associated special arrival and departure timestamps, the additional latency added by this slow path would be indistinguishable from actual network latency

4.3.1.1 IP SLA Network Management Support

IP SLA, described in detail in Chapter 8, can be managed by third party tools such as NetVoyant from NetQoS. IP SLA has a very strong SNMP-based configuration and data collection interface, and NetVoyant offers an easy GUI for managing Simple Network Management Protocol (SNMP) devices using a central console, rather than managing each device individually. The MIB browser in the NetVoyant console supports direct access to the MIB tables of a device.

4.3.1.2 IP SLA Operations

There are several key IP SLA operations:

- Internet Control Message Protocol (ICMP) echo
- User Datagram Protocol (UDP) echo
- Domain Name System (DNS) request

- Hypertext Transfer Protocol (HTTP) requests

4.3.1.3 IP SLA Configuration

This section provides configuration examples.

4.3.1.3.1 General Configuration Commands:

```
Router(config)#:ip sla <operation number>
```

Begin configuration for an ip sla operation and enter IP SLA monitor mode.

```
Router(config)#:ip sla monitor schedule <operation number> <start-time><age out>
<recurrence>
```

Configure the scheduling parameters for an individual IP SLA. This command must be run before an IP SLA will begin.

4.3.1.3.2 General Show Commands

```
Router#sh ip sla configuration <operation number>
```

This example shows the configuration parameters set for the current IP SLA by the specified operation number.

Example

```
Router#sh ip sla configuration 1
IP SLAs, Infrastructure Engine-II.
Entry number: 3
Owner: ICMP Echo - 100.1.1.161 - 60.1.1.100
Tag: WANOPT ICMP ECHO
Type of operation to perform: icmp-echo
Target address/Source address: 60.1.1.100/0.0.0.0
Operation timeout (milliseconds): 5000
Type Of Service parameters: 0x0
Vrf Name:
Request size (ARR data portion): 28
Verify data: No
Schedule:
   Operation frequency (seconds): 60  (not considered if randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): Forever
   Entry Ageout (seconds): 3600
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 4294967295
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None
Enhanced History:
Router#sh ip sla statistics 1
```

This command shows basic statistics gathered by the specified IP SLA .

Example

```
Router#sh ip sla statistics 1
Round Trip Time (RTT) for       Index 1
        Latest RTT: 60 milliseconds
Latest operation start time: 15:18:20.255 EST Tue Dec 4 2007
Latest operation return code: OK
Number of successes: 58
Number of failures: 0
       Operation time to live: Forever
```

4.3.1.3.3 Configuring ICMP Echo

```
Router#: conf t
Router(config)#:ip sla 1
Router(config-ip-sla)#: icmp-echo 52.1.1.100
Router(config-ip-sla)#ip sla schedule 1 start-time now life forever
Router(config-ip-sla)#:end
Router#:wr
```

4.3.1.3.4 Configuring UDP Echo

```
Router#: conf t
Router(config)#:ip sla 1
Router(config-ip-sla)#:udp-echo 52.1.1.100 443
Router(config-ip-sla)#ip sla schedule 1 start-time now life forever
Router(config-ip-sla)#:end
Router#:wr
```

4.3.1.3.5 Configuring DNS Echo:

```
Router#: conf t
Router(config)#:ip sla 1
Router(config-ip-sla)#:dns www.cisco.com name-server 52.1.1.100
Router(config-ip-sla)#ip sla schedule 1 start-time now life forever
Router(config-ip-sla)#:end
Router#:wr
```

4.3.1.3.6 Configuring HTTP Echo:

```
Setup:
Router#: conf t
Router(config)#:ip name-server 52.1.1.100
Router(config)#:ip domain-list wanopt4.cisco.com
Router(config)#:ip domain-name wanopt4.cisco.com
Router(config)#:exit
Router#:wr
Router# conf t
Router(config)# ip sla 1
Router(config-ip-sla)#http get url http://www.cisco.com
Router(config-ip-sla)#ip sla schedule 1 start-time now life forever
```

4.3.1.4 ICMP Echo

ICMP is usually the first tool used in network troubleshooting to verify connectivity between two points on the network. With ICMP echo, several ICMP echo packets are sent to a destination, which then responds with the ICMP echo-replies. A bidirectional check such as ICMP echo can quickly verify connectivity to the target device.

ICMP echo operation can also monitor end-to-end response time between a Cisco router and a network resource or IP host. To compute response time, the time between sending an ICMP echo request and receiving an ICMP echo reply is measured. Only complete transaction round-trip time (RTT) is measured.

4.3.1.5 UDP Echo

UDP echo can determine round-trip delay times for UDP packets and test connectivity to both Cisco and non-Cisco devices, which can be very useful in troubleshooting certain business-critical applications. UDP echo offers more detailed reporting than ICMP operations, such as one-way delay measurement, but only when used with the IP SLA responder.

4.3.1.6 DNS Request

DNS is commonly used to translate hostnames into IP addresses, and to translate IP addresses into hostnames. DNS request measures the amount of time it takes to send a DNS request to a DNS server and receive a response to the request. This request can contain either an IP address or hostname depending on which is specified when the SLA is setup. DNS operations are a critical element for determining a network's overall performance as most IP services depend heavily on DNS name resolution.

4.3.1.7 HTTP Operation

HTTP operation centers around monitoring the response time between the source device and the HTTP server. Three values are measured to calculate response time:

1. DNS lookup: Round Trip Time (RTT) of a DNS lookup
2. TCP Connect: RTT of a TCP connect to the HTTP server
3. HTTP Transaction Time: RTT taken from request to response from the HTTP server

The HTTP SLA has two requests that can be configured: HTTP Get and HTTP RAW. For HTTP Get requests, the IP SLA formats the request based on the specified URL. For RAW requests, the entire content of the HTTP request must be specified. This allows RAW requests control over fields such as authentication.

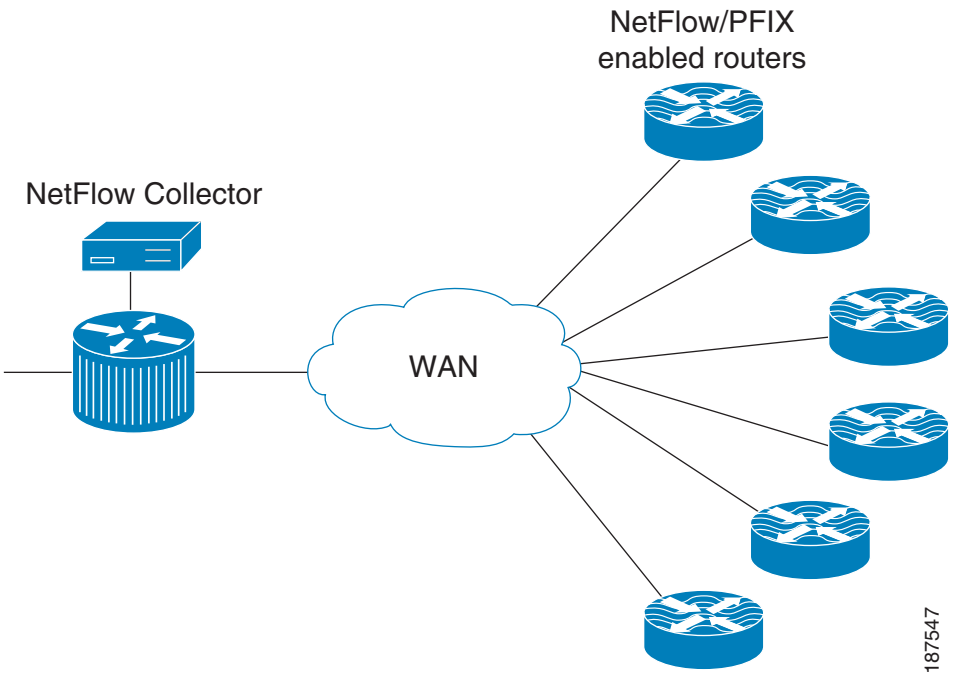
4.3.2 NetFlow

Cisco IOS NetFlow is an integral technology in IOS network statistics gathering. NetFlow collects packets, maps them into flows, and counts the collected flow statistics as the packets enter and exit an interface. These flow statistics can then be exported to a NetFlow collector for storage and analysis.

The key components of NetFlow are the cache stores that hold IP flow information and an export mechanism that can send NetFlow data to a remote collector such as the Cisco NetFlow Collector. Figure 4-2 shows a NetFlow Collector.

NetFlow operates by creating a NetFlow cache entry for each active flow and maintains a separate flow record within the cache for active flows. Each of these flow records contain multiple data fields which themselves are exported to the NetFlow Collector.

Figure 4-2 NetFlow Collector



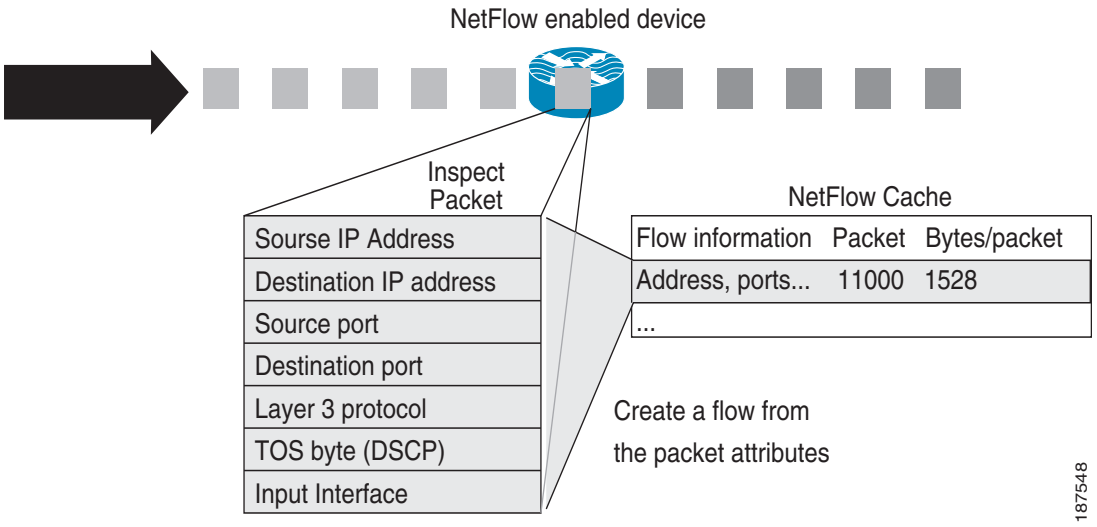
NetFlow identifies packet flows for IP Packets by looking at a number of fields in the data packet. A flow is defined as a set of packets having common properties. NetFlow defines a flow as the combination of the following seven key-fields, which determine how a flow is identified:

1. Source IP Address
2. Destination IP Address
3. Source port number
4. Destination port number
5. Layer 3 protocol type (e.g., ICMP, TCP, UDP)
6. ToS byte
7. Logical input interface (ifIndex)

Each flow record is created by grouping packets with the same characteristics into a flow. This method of determining a flow is ideal because a large amount of network information is condensed into a database of NetFlow information called the NetFlow cache. If any of these fields are different from another flow, it is considered a different flow.

NetFlow operates by creating a NetFlow cache entry that contains information for each active flow., as illustrated in Figure 4-3.

Figure 4-3 NetFlow Cache Entry



4.3.2.1 NetFlow Cache

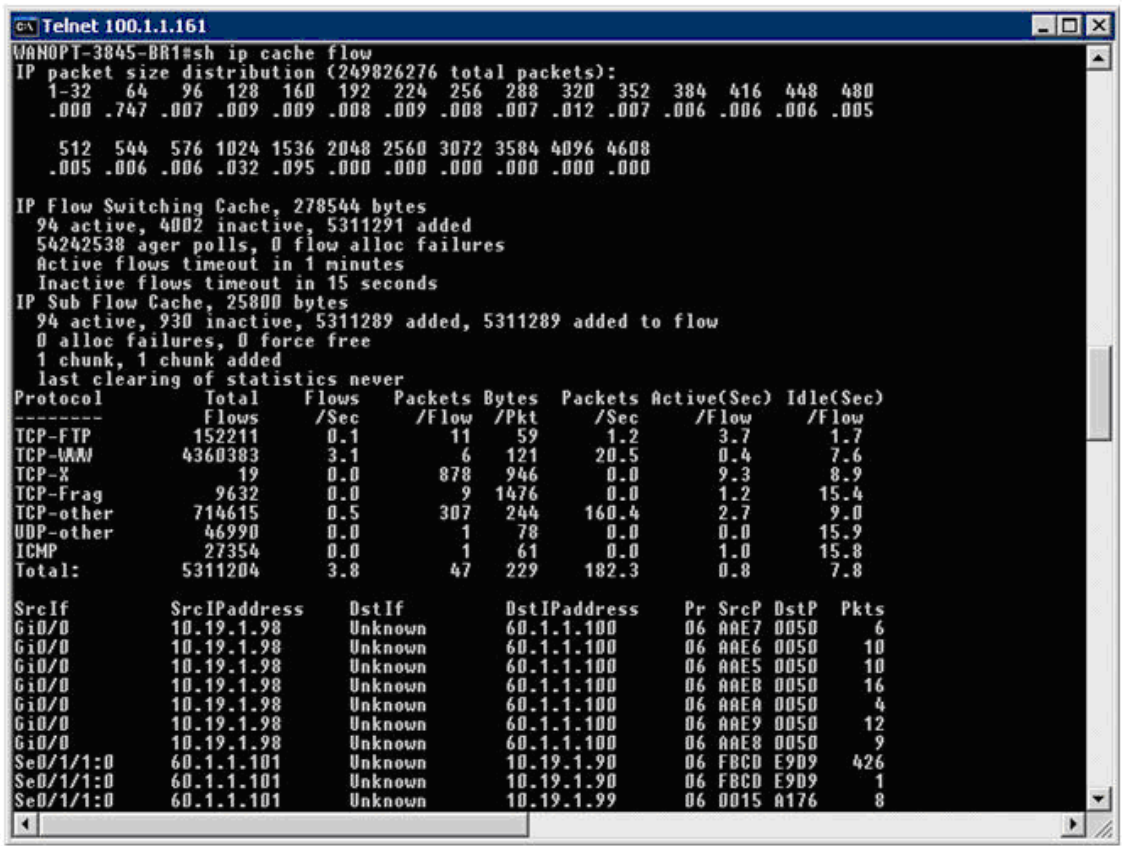
The attributes of active flows can be analyzed by displaying the NetFlow Cache. This makes NetFlow a powerful troubleshooting tool, even without flow exporting.

4.3.2.2 Show Command

```
Router#: sh ip cache flow
```

Figure 4-4 illustrates NetFlow cache entries.

Figure 4-4 NetFlow Cache Entries



The various segments comprise:

- Packet size distribution
- General statistics about the state of the NetFlow cache
- The time a particular flow remains active in the cache before it is discarded.
- Flow breakdown by some well known protocols
- Actual NetFlow Cache entries

4.3.2.3 Aging Flows

On the NetFlow accounting device, the rules for expiring flow records and exporting them from cache entries to a flow collector are the following:

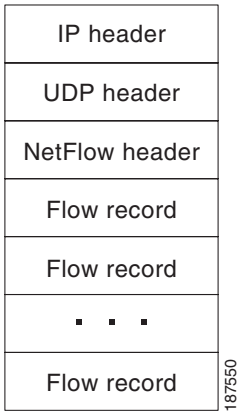
- Inactive/Active Timer: Flows that have been idle for a specified time are expired and removed. The default setting for this timer is fifteen seconds of traffic inactivity and can be configured between 10 and 600 seconds. On the other hand, long-lived flows are also expired and removed from the cache based on a different timer, called the active timer. The cutoff time for active flow expiration is thirty minutes and can be configured between 1 and 60 minutes.
- Full Cache: If a cache approaches full, emergency expiration will occur. The cache size can be configured by the network operator.
- End of a TCP connection: TCP connections at the end of a byte stream (FIN) or have been reset (RST) automatically expire on software platforms.

4.3.2.4 NetFlow Versions

4.3.2.4.1 NetFlow Version 5

NetFlow Version 5, generally considered the most popular NetFlow version, includes BGP Autonomous System information and flow sequence numbers.

Figure 4-5 Typical NetFlow Export Datagram Format for Versions 1, 5, 7, and 8



4.3.2.4.2 Configuration Commands

Set flow source destination as local device:

```
Router(config)#: ip flow-export source loopback 0
```

Set NetFlow export version:

```
Router(config)#: ip flow-export version 5 peer-as
```

Specify the NetFlow collector for exported records:

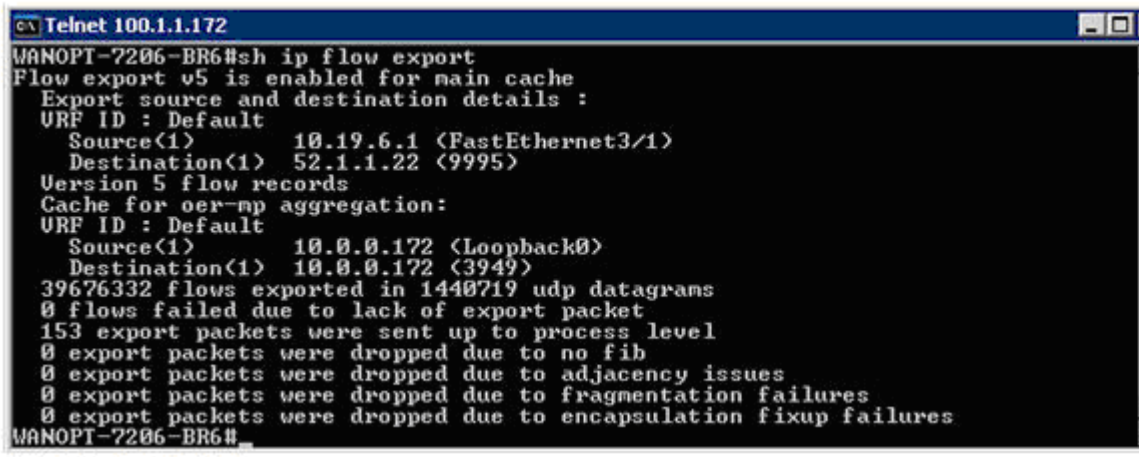
```
Router(config)#: ip flow-export destination 52.1.1.22 9995
```

4.3.2.4.3 Show Commands:

This command displays the NetFlow Version 5 configurations as well as other interesting statistics, such as the number of exported flow records, the number of exported packets, the number of packets that were not exported, and the reason for failures.

```
Router#: sh ip flow export
```

Figure 4-6 IP Flow Export Statistics



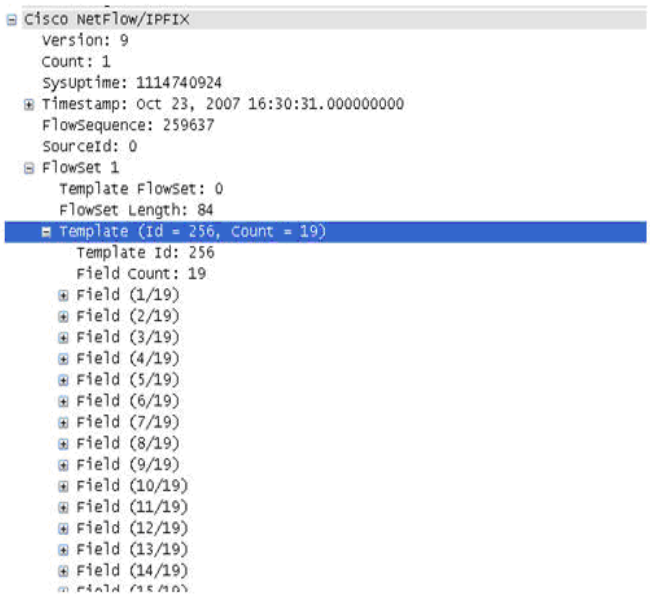
4.3.2.4.4 NetFlow Version 9

The distinguishing feature of the NetFlow Version 9 format is that it is *template based*. Templates provide an extensible design to the record format, a feature that should allow future enhancements to NetFlow services without requiring concurrent changes to the basic flow-record format and collector code. Version 9 also incorporates new features such as multicast, MPLS, BGP next hop, and IPv6. Using templates with NetFlow Version 9 provides several key benefits:

- Third-party business partners who produce applications that provide collector or display services for NetFlow will not be required to recompile their applications each time a new NetFlow export field is added. Instead, they may be able to use an external data file that documents the known template formats.
- New features can be added to NetFlow more quickly, without breaking current implementations.
- NetFlow is considered "future-proofed" against new or developing protocols, because the Version 9 format can be adapted to provide support for them and other non-Flow based data measurements.

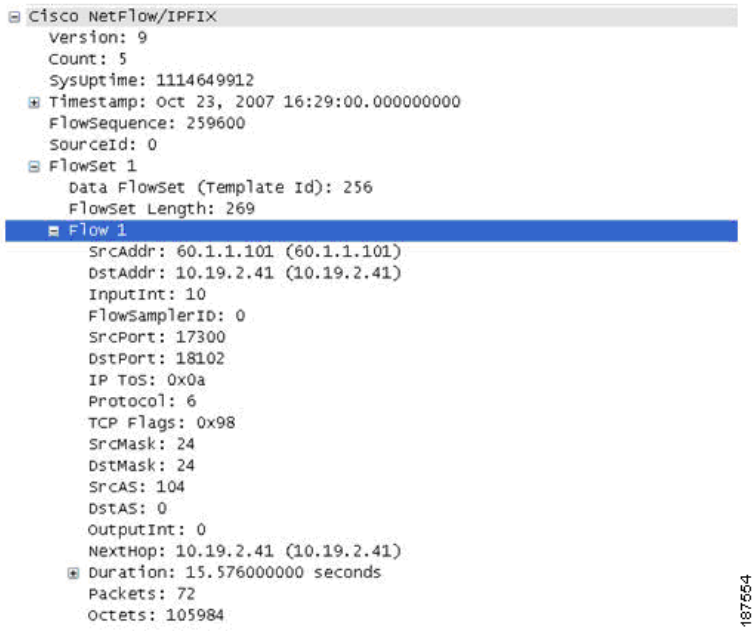
In the following NetFlow version 9 flow captured using WireSharks, Figure 4-7 depicts the NetFlow v9 flow template that identifies the fields that will be present in the actual flow record, while Figure 4-8 depicts the actual flow record.

Figure 4-7 NetFlow version 9 Flow Template



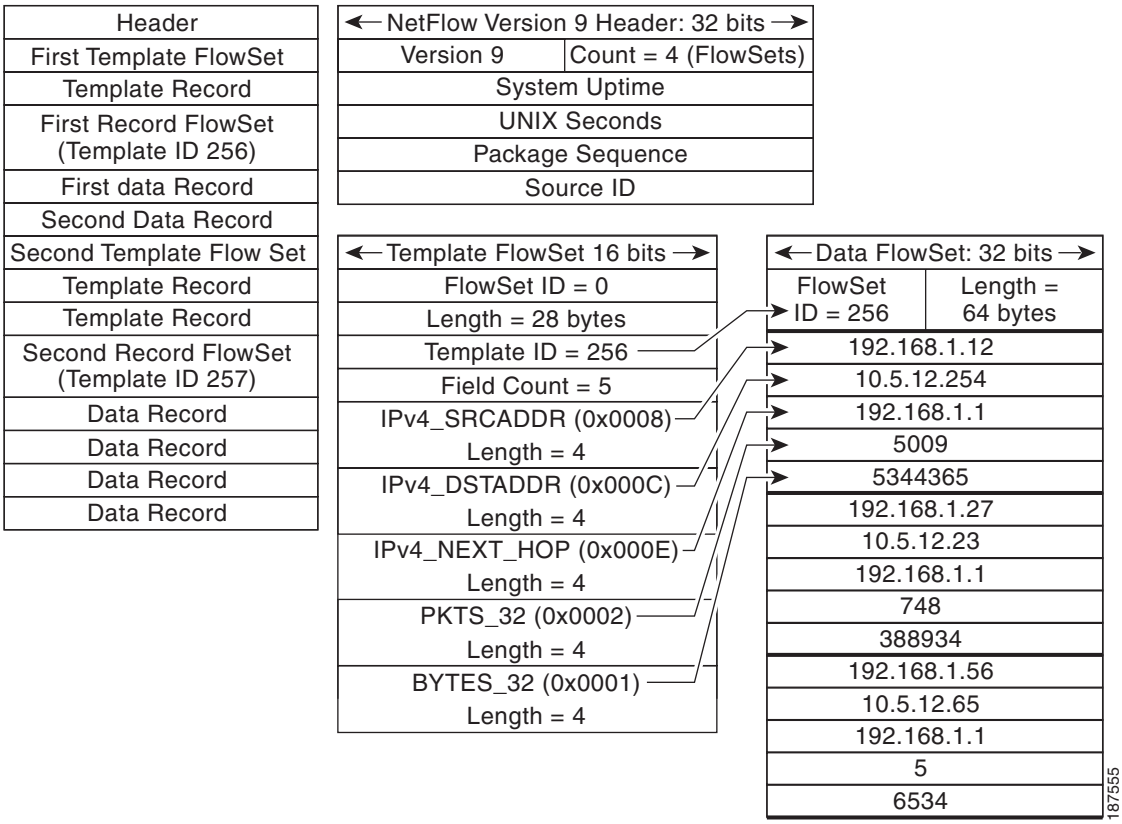
Highlighted here is the flow template that shows the template ID for this flow record and the number and type of fields included in the record.

Figure 4-8 NetFlow version 9 Flow Record



A more detailed view of how the NetFlow Version 9 flow template and flowsets match up is depicted in Figure 4-9.

Figure 4-9 NetFlow version 9 Flow Breakdown



4.3.2.4.5 Configuration Command

```
Router#:conf t
Router(config)#: ip flow-export version 9
Router(config)#: ip flow-export destination  52.1.1.22 9995
Router(config)#: ip flow-export source loopback 0
```

4.3.2.4.6 Show Commands

Same as netflow version 5:

```
Router#: show ip flow export
```

This command shows the number of templates and the number of active templates.

```
Router#: show ip flow export template
```

4.3.3 NBAR

Network Based Application Recognition (NBAR) provides network traffic classification. NBAR can recognize a very wide variety of applications by doing IP packet inspection up to OSI Layer 7. It can, for instance, differentiate between Web-based HTTP and Skype traffic, which can both use TCP port 80.

When an application is recognized, NBAR classifies the traffic for performance and accounting purposes. This function gives an operator the ability to invoke any range of services for that specific application, whether offering more or less bandwidth, latency queuing, or completely blocking certain packets.

NBAR also provides a special Protocol Discovery (PD) feature that determines which applications and protocols are traversing the network at any given time. PD captures key statistics that are associated with each protocol based on IP flows. Like Cisco NetFlow, NBAR defines IP flows as a unidirectional flow of IP packets that share the following five values:

- Source IP address
- Destination IP address
- Source port
- Destination port
- L3 protocol type

NetFlow and NBAR both leverage L3 and L4 header information. However, unlike NetFlow, NBAR also examines data from L3–L7. NBAR uses L3 and L4 and packet inspection for classification, and supports stateful inspection of dynamic-port traffic. NBAR also requires a set number of packets before making a protocol distinction.

4.3.3.1 NBAR Protocol Discovery

NBAR PD provides an easy way to discover application traffic flowing through an interface by providing a Protocol Discovery (PD), MIB, which gives it expanded capabilities through SNMP. This includes:

- Enable/Disable Protocol Discovery on a per interface basis
- Monitoring both ingress and egress traffic
- Display statistics on a per-protocol basis

Table 4-1 NBAR Protocol Discovery MIB Details

Table	Description
cnpdSupportedProtocols	List of all supported protocols NBAR supports
cnpdAllStats	All NBAR statistics per interface such as Packet counters (inbound/outbound) Byte counters (inbound/outbound) Bit rate (inbound/outbound)
cnpdTopNStats	Top-N table statistics
cnpdThresholdhistory	History of falling or rising events
cnpdStatus	Enable or disable NBAR per interface, including time stamp
cnpdTopNConfig	Configure the Top-N table by interface
cnpdThresholdConfig	Protocol threshold configuration
cnpdNotificationsConfig	Enable traps

cnpdMIBNotifications	Rising or falling events

4.3.3.2 Configuration

4.3.3.2.1 Configuring Protocol Discovery on a Specific Interface

```
router#: conf t
router(config)#: interface gigabitEthernet 0/1
```

This configures NBAR to discover traffic and keep traffic statistics for all protocols known to NBAR on a particular interface

```
router(config-if)#:ip nbar protocol-discovery
router(config)#: exit
```

Optional

This configures NBAR to search for a protocol or protocol name using port number(s) other than an already known port. Up to 16 different port numbers can represent a protocol.

```
router(config)#: ip nbar port-map
```

Configures NBAR to classify and monitor additional static port applications:

```
router(config)#:ip nbar custom protocol-name [destination | source] [tcp|udp]
```

The parameters are:

- protocol-name: Specifies the name of the user defined protocol.
- number: The byte location of the value to be searched in the payload (0 to 255)(optional).
- destination: Inspects destination flows only (optional)
- tcp: Defines up to 16 explicit TCP port numbers or a range of a maximum of 1000 TCP ports
- udp: Defines up to 16 explicit UDP port numbers or a range of a maximum of 1000 UDP ports.

Extends the list of protocols by loading a new PDLM (providing the full path to the PDLM) New PDLM versions are provided on the Cisco website at http://www.cisco.com/go/nbar:

```
router(config)#:ip nbar pdlm
```

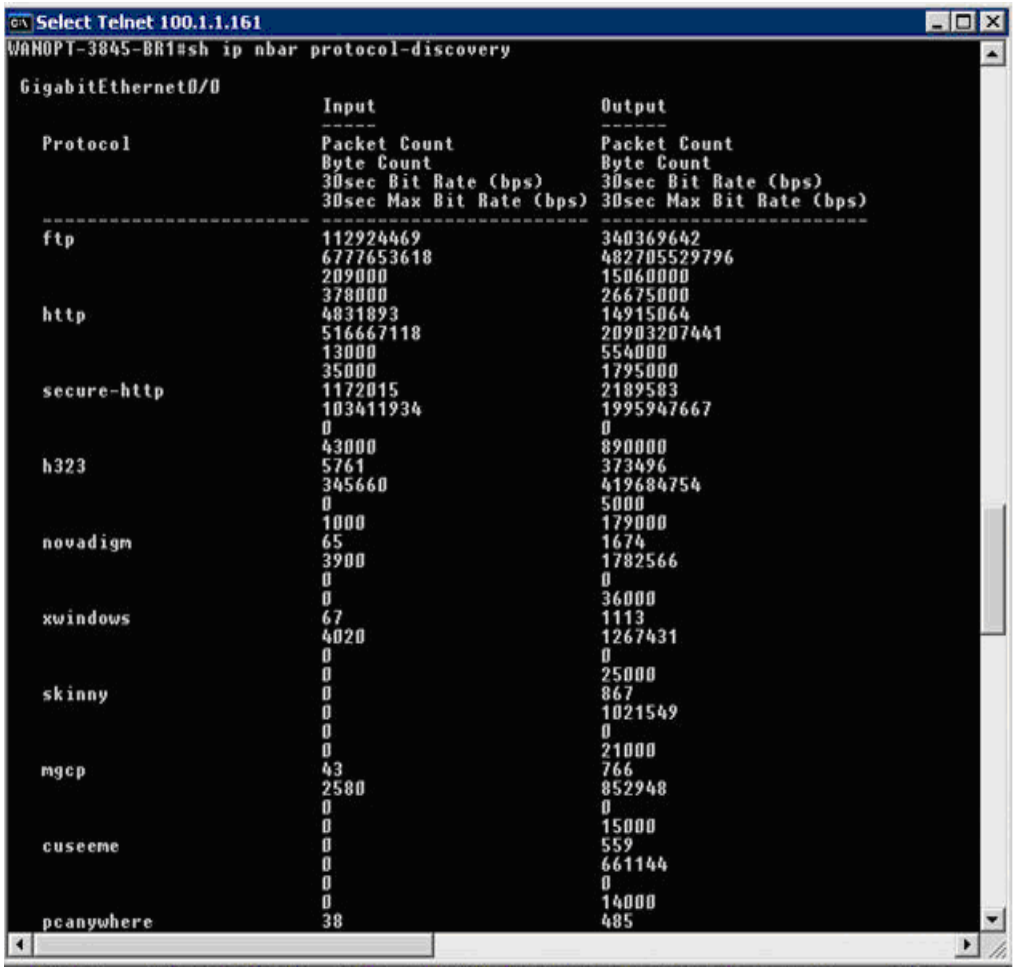
4.3.3.3 Show Commands

```
router#: show ip nbar protocol-discovery
```

This command displays the statistics gathered by the NBAR Protocol Discovery feature. By default, all statistics for all interfaces are displayed.

Note Egress Traffic statistics are gathered before policing features.

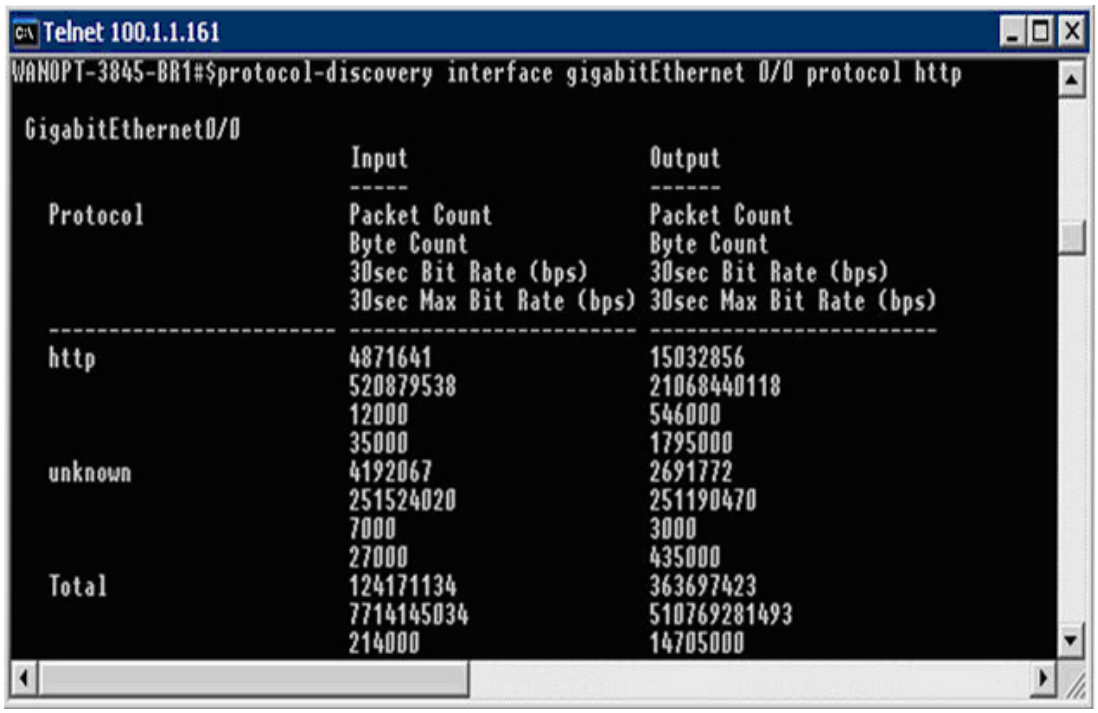
Figure 4-10 Sample Output from PD Show Command




```
Router#: show ip nbar protocol-discovery
```
interface stats protocol top-n

- interface-spec: specifies an interface to display
- stats: specifies the byte count, bit rate, or packet count is to be displayed
- protocol: specifies that statistics for a specific protocol.
- top-n: specifies that a Top-N of most active protocols is displayed

Figure 4-11 Figure 4-11. Sample Output from PD Interface Show Command



4.3.4 CBQoS MIB

The Cisco Class-Based QoS (CBQoS) MIB supplies QoS information for Cisco network elements that support the Modular QoS command-line interface (MQC). CBQoS provides configuration capabilities and monitoring statistics that include summary counts and rates by traffic class before and after the enforcement of QoS policies. It also provides detailed feature-specific statistics that are available for select PolicyMap features. Policy actions are defined per interface and traffic direction, whether ingress or egress. The CBQoS MIB supports both 32 bit and 64 bit counters.

The following is a list of relevant MIB tables for QoS and contain only statistical information.

- **cbQosClassMapStats-** Statistical information about class maps, such as pre/post-policy packet/byte counts, bit rates, drop packet/bytes and no-buffer drops.
- **cbQosMatchStmtStats-** Statistical information about match statement-specific information, such as prepolicy packet/byte
- **cbQosPoliceStats-** Statistical information about police actions, such as conformed or exceeded packet/byte counters and bit rates.
- **cbQueueingStats-** Statistical information about queuing actions, such as the various queue depth and discar packet/byte counters.
- **cbQosTSStats-** Statistical information about traffic-shaping actions, such as various delay and drop packet/byte counters, state of feature, and queue size.
- **cbQosREDClassStats-** Statistical information about per-precedence weighted random early detection actions, such as random packet/byte counters and tail drop packet/byte counters.
- **cbQosPoliceActionCfg-** Required objects to display class-based QoS objects' configuration information.

4.4 Additional Instrumentation

Additional instrumentation includes:

- Cisco WAAS Flow Agent
- Connection State and Operation Statistics Reports

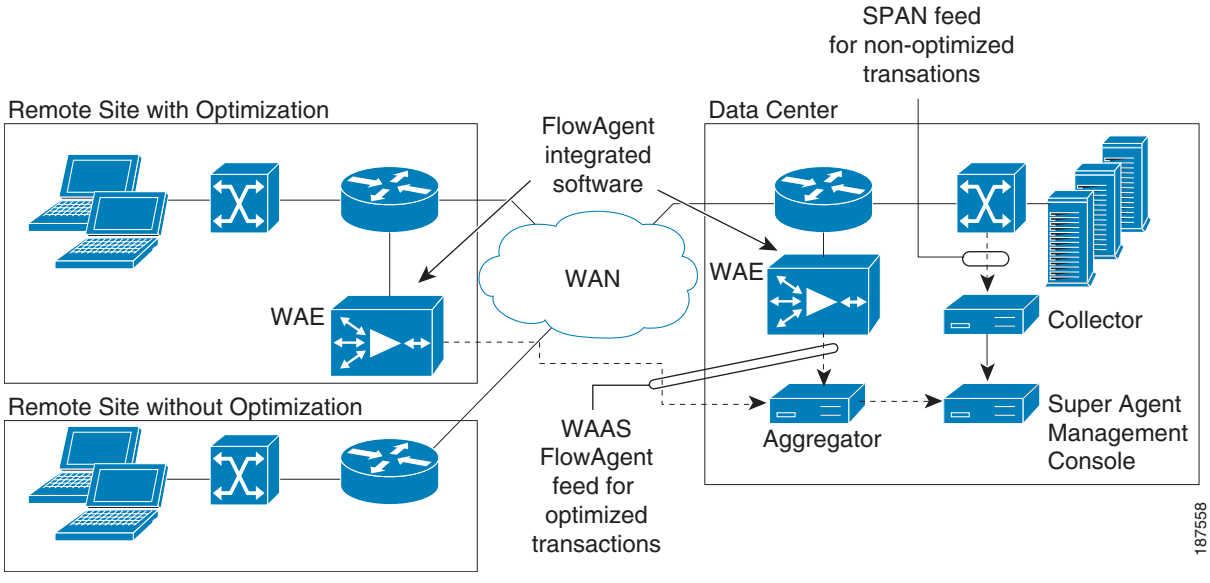
4.4.1 Cisco WAAS Flow Agent

Cisco and NetQoS jointly developed monitoring instrumentation to allow accurate end-to-end response time measurements, from the client to the server, over links optimized by Cisco WAAS devices. Central to this instrumentation is a Cisco software feature called the FlowAgent, a flow monitoring module integrated in the WAE. The FlowAgent captures important packet information and sends it across the network to a third-party monitoring agent (for example, NetQoS SuperAgent). It exports all data necessary to report application response times experienced at remote sites served by WAAS and the detailed performance metrics for each optimized link.

The FlowAgent, illustrated in Figure 4-12, is part of the standard software image for Cisco WAAS 4.0.13 and newer. When configured on a WAE, the FlowAgent collects relevant flow information for optimized TCP transactions and transmits it to the SuperAgent Aggregator, a device that is dedicated solely to FlowAgent data collection.

The SuperAgent Aggregator processes the flow information and sends it to a SuperAgent Management Console for storage and reporting. FlowAgent data collection can be configured on the Cisco WAAS Central Manager selectively for a single WAE device, or for multiple (or all) WAE devices using device groups.

Figure 4-12 Cisco WAAS FlowAgent



4.4.1.1 Sample Export of Flow Records for Optimized Traffic

The FlowAgent captures the following information on optimized traffic and sends it over to the configured SuperAgent Aggregator:

- Source IP address
- Destination IP address
- Source TCP port
- Destination TCP port
- TCP Sequence number
- TCP Acknowledgement number
- TCP payload byte count
- Packet arrival time in milliseconds
- IP identifier
- WAE MAC address Src/Dst flag
- TCP flags

4.4.1.2 FlowAgent Configuration

The SuperAgent FlowAgent monitoring agent is composed of two modules: the console (or host) and the collector. The console IP address is configured on the WAE through the WAE CLI or through the Central Manager GUI. The WAE initiates a temporary connection to the monitoring agent console. This temporary connection is referred to as the control connection. The control connection uses TCP port 7878, and its purpose is to obtain the IP address and port number of the collector to which the WAE is assigned. The WAE also pulls the configuration information regarding which servers are to be monitored over the control connection. Once the WAE obtains the IP address and port number information of the collector, the WAE opens a persistent connection to the collector. Collected summary data for the servers that are being monitored is sent over this persistent connection. The WAE send summary data only to Aggregators for servers assigned to it.

Configuration for flow monitoring with NetQoS involves the following tasks:

1. From the WAE CLI or Central Manager GUI, enter the SuperAgent Master Console IP address in the **tcpstat-v1** Host field on your WAE appliances.

   If you are configuring multiple appliances through a device group, wait for the configuration to propagate to all the appliances in the device list.

2. From the NetQoS SuperAgent console, assign a WAE to a SuperAgent Aggregator (known as the collector in WAAS terminology) and configure the NetQoS Networks, Servers, and Applications entities.

4.4.1.3 Enabling the FlowAgent on the WAE using the Central Manager GUI

To configure flow monitoring on your WAEs using the Central Manager GUI, follow these steps:

1. From the NetQoS SuperAgent console, assign a WAE to a SuperAgent Aggregator (known as the collector in WAAS terminology) and configure the NetQoS Networks, Servers, and Applications entities.
2. Create a new device group to be used for configuring flow monitoring on multiple devices. To create a device group, choose **Devices > Device Groups > Create New Device Group**.

When you create the device group, check the auto assign all newly activated devices to this group check box to enable this option.

3. Add your existing WAE devices to this new device group.
4. From the Device Group listing window, click the **Edit** icon next to the name of the flow monitoring configuration device group that you want to configure.
5. In the Contents pane, choose **General Settings > Notification and Tracking > Flow Monitor**. The Flow Monitor Settings for Device Group window appears.
6. Check the **Enable** check box.
7. In the tcpstat-v1 Host field, enter the IP address of the monitoring agent console.

   This configuration enables the WAE to establish a temporary connection (a control connection) to the console for obtaining the IP address of the SuperAgent aggregator. You must configure the aggregator IP address information from the console device. (See the configuration documentation for your third-party flow monitoring application software.)
8. Click **Submit** to apply the settings to the devices in this device group.

4.4.1.4 Enabling the FlowAgent on the WAE using the CLI

To configure flow monitoring on your WAEs using the CLI, two commands are required (Figure 4-13):

1. Register the WAE with the IP address of the monitoring agent console by using the **flow monitor tcpstat-v1 host** global configuration command. The WAE needs to know the location of the SuperAgent Management Console

   ```
   #flow monitor tcpstat-v1 host <IP_Address>
   ```
   This configuration allows the WAE to establish a temporary connection (a control connection) to the console (or host) for obtaining the IP address of the collector. You must configure the collector IP address information from the console device. (See the configuration documentation for your third-party flow monitoring application software.)

   Note The IP Address in this command is the SuperAgent Management Console. The SuperAgent Aggregator IP address is not needed for this configuration since the SuperAgent Aggregator to WAE configuration is accomplished in the SuperAgent Management Console UI.
2. Enable flow monitoring on the WAE appliance by using the **flow monitor tcpstat-v1 enable** global configuration command.

   ```
   #flow monitor tcpstat-v1 enable
   ```

   Note WAE requires the IP Address of the SuperAgent Management Console before using the enable command.

Figure 4-13 Enabling FlowAgent on the WAE

4.4.2 Connection State and Operation Statistics Reports

The console (or host) module and the collector module may be on a single device or may be located on separate devices. These connections are independent of one another.

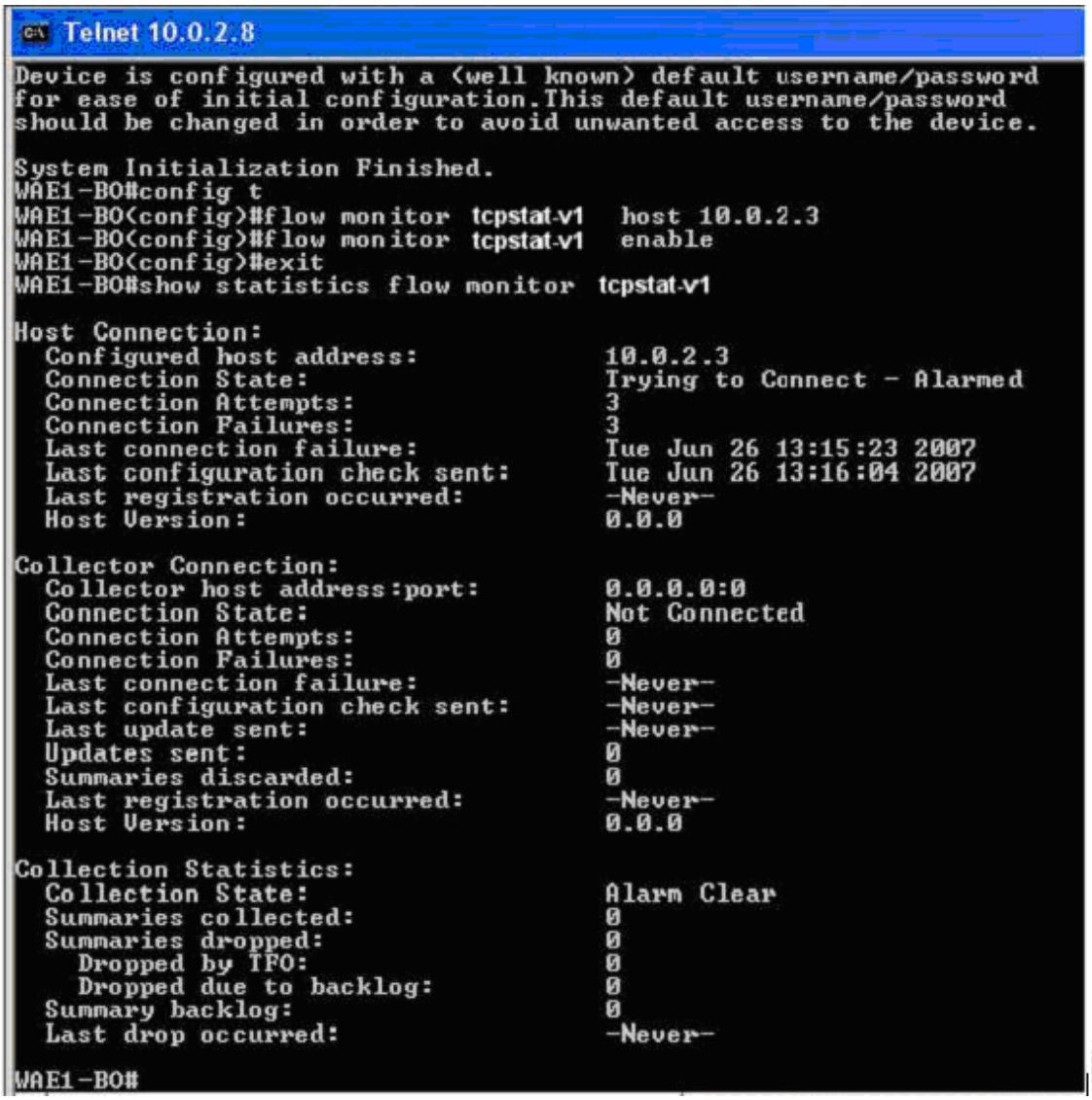
The state of these connections, as well as various operation statistics, is reported by the **show statistics flow monitor tcpstat-v1** EXEC mode command. This command is run from the WAE to determine the status of its connection to the SuperAgent system. As shown in Figure 4-14, this command provides the following information:

- Host Connection shows the IP of the SuperAgent Management Console. It usually shows "Not Connected" unless running it during a WAE to Console poll.
- The connection between the SuperAgent Management Console and WAE is non-persistent. It will reconnect and synchronize every 5 minutes).
- Collector Connection should always show "Connected" after the IP:Port of the assigned SuperAgent Aggregator for that WAE.
- The assigned Aggregator is setup in the SuperAgent GUI. You will probably have one Aggregator even in a large network. It acts more like a Harvester than a Collector.

Figure 4-14 FlowAgent Connection Status

Figure 4-15 illustrates a WAE that has not connected to the SuperAgent Management Console and has never received SuperAgent Aggregator assignment. An Alarm state will be raised in the State field.

Figure 4-15 FlowAgent connection status failure



Connection errors and data transfer errors raise alarms on the WAE and in the Central Manager GUI. For debug information, use the **debug flow monitor tcpstat-v1** EXEC mode command.

The **`show statistics flow filters`** command identifies which filters were built, based on the server assignment from the SuperAgent Management Console. In Figure 4-16, only the 10.0.3.6 server traffic is being optimized as shown in the "flow hits" column. Note that the server list comes from the SuperAgent Management Console automatically. This is the best way to validate the SuperAgent Management Console configuration against the WAE if data does not appear in the graphs.

Figure 4-16 Identifying Built Filters from the SuperAgent Management Console

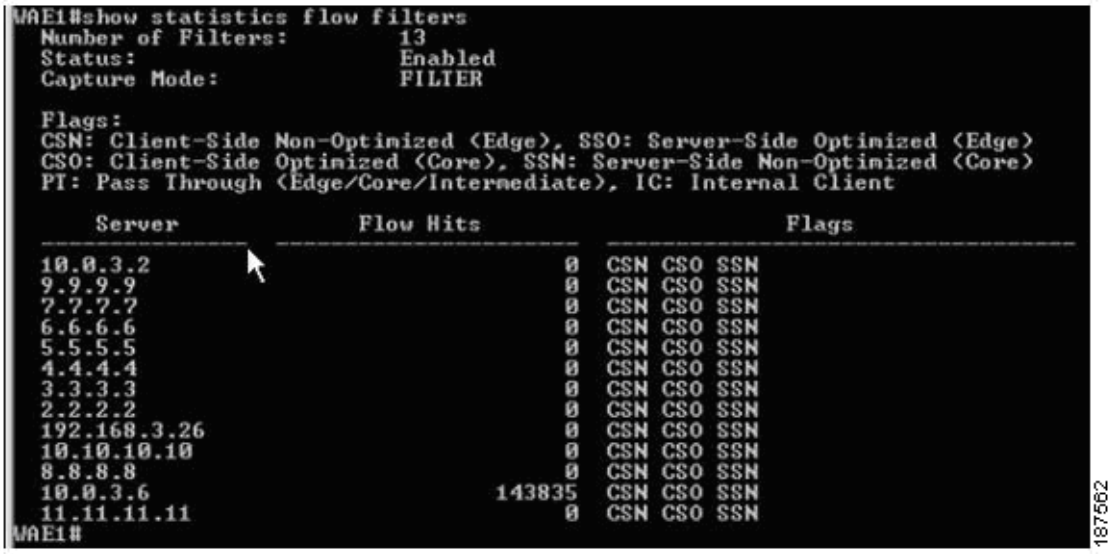
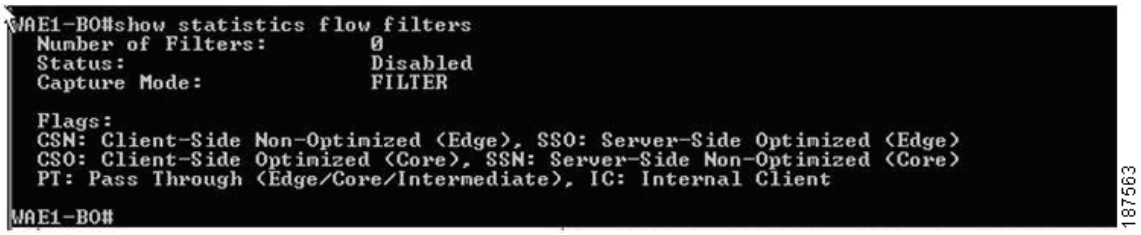


Figure 4-17 depicts either a communication issue with the console or that there are no servers defined in SuperAgent. To resolve, ensure that servers are configured within the SuperAgent Management Console and use the **show statistics flow monitor tcpstat-v1** command to troubleshoot connectivity problems.

Figure 4-17 Problem Reported in the SuperAgent Management Console



4.5 Summary

Cisco IOS measurement tools give IT directors and managers the needed tools to benchmark the various components of their network. Cisco IOS comes packaged with tools such as NetFlow and IPSLA polling that provide the platform to build detailed network monitoring abilities as well as the WAAS Flow Agent that allows network visibility into optimized TCP traffic. This visibility can reduce the amount of time required to troubleshoot network issues and bring about a resolution as cost effectively as possible.