Introduction to Cisco IOS® Flexible NetFlow

Last updated: September 2008

The next-generation in flow technology allowing optimization of the network infrastructure, reducing operation costs, improving capacity planning and security incident detection with increased flexibility and scalability.

Introduction

NetFlow invented by Cisco has become the standard for acquiring IP operational data for many customers. Visibility into the network is an indispensable tool. In response to new requirements and pressures, network operators are finding it critical to understand how the network is behaving including:

- Application and network usage
- Understand who, what, when, where, and how network traffic is flowing
- Network efficiency and utilization of network resources
- The impact of changes to the network
- Network anomaly and security vulnerabilities
- Long term compliance, business process and audit trail

Applications for NetFlow data are constantly being invented but the key usages include:

- Real-time network monitoring
- Application and user profiling
- Network planning and capacity planning
- Security incident detection and classification
- Accounting and billing
- Network data warehousing, forensics and data mining
- Troubleshooting

NetFlow Based Network Awareness

The ability to characterize IP traffic and understand who sent it, the traffic destination, the time of day, the application information, is critical for network availability, performance and troubleshooting. Monitoring IP traffic flows facilitates more accurate capacity planning and ensures that resources are used appropriately in support of organizational goals. It helps Cisco customers determine how to optimize resource usage, plan network capacity, where to apply Quality of Service (QoS) and it plays a vital role in network security to detect Denial–of–Service (DoS) attacks and network-propagated worms.

NetFlow facilitates solutions to many common problems encountered by network professionals as shown in Table 1.

Table 1. Common Solutions Facilitated by NetFlow

NetFlow Facilities Solutions To:	Description
Analyze new applications and their network impact	Identify new application network load such as VoIP or remote site additions.
Reduction in peak WAN traffic	Use NetFlow statistics to measure WAN traffic improvement from application-policy changes; understand who is utilizing the network and the network top talkers.
Troubleshooting and understanding network pain points	Diagnose slow network performance, bandwidth hogs and bandwidth utilization in real-time with command line interface or reporting tools.
Detection of unauthorized WAN traffic	Avoid costly upgrades by identifying the applications causing congestion.
Security and anomaly detection	NetFlow can be used for anomaly detection, worm diagnosis along with applications.
Validation of QoS parameters	Confirm that appropriate bandwidth has been allocated to each Class of Service (CoS) and that no CoS is over- or under-subscribed.

For a primer in the basics of NetFlow please read the “Introduction to NetFlow—A Technical Overview” document first.

The Next Generation in Flow Technology—Flexible NetFlow

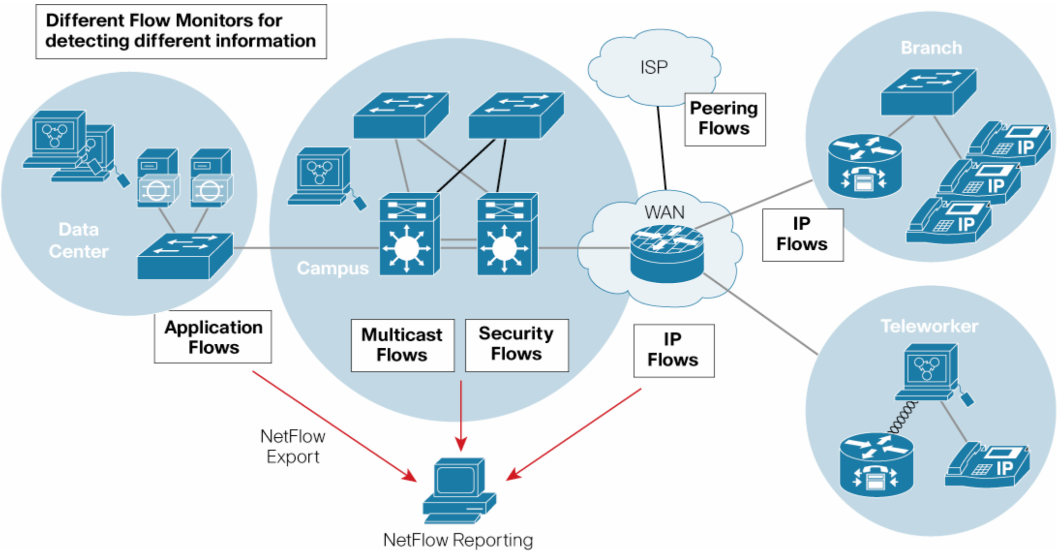
Cisco is now innovating flow technology to a new level beyond what has been traditionally available. *Cisco IOS Flexible NetFlow* is Cisco’s next-generation flow technology. Flexible NetFlow provides enhanced optimization of the network infrastructure, reduces costs, and improves capacity planning and security detection beyond other flow based technologies available today.

Key Advantages to using Flexible NetFlow:

- Flexibility, scalability, aggregation of flow data beyond traditional NetFlow
- The ability to monitor a wider range of packet information producing new information about network behavior
- Enhanced network anomaly and security detection
- User configurable flow information to perform customized traffic identification and the ability to focus and monitor specific network behavior
- Convergence of multiple accounting technologies into one accounting mechanism

It provides a NetFlow architecture that can track multiple NetFlow applications simultaneously. For example, the user can create simultaneous and separate Flow Monitors for security analysis and traffic analysis. Cisco IOS Flexible NetFlow provides enhanced security detection and or network troubleshooting by allowing customization of flow information. For example, the user can create a specific Flow Monitor to focus and analyze a particular network issue or incident. It provides real-time monitoring with immediate flow cache capabilities and long term or permanent tracking of flow data. Cisco IOS Flexible NetFlow will enhance NetFlow’s already rich feature capabilities allowing the tracking of information at layer 2 for switching environments, layer 3 and 4 for IP information and up to layer 7 with deep packet inspection for application monitoring. Figure 1 shows various types of Flow Monitors to view and understand network behavior.

Figure 1. Example of Flexible NetFlow Customizable Flow Monitors



An Example of Application Tracking with Flexible NetFlow

Flexible NetFlow unlike traditional NetFlow allows the user to customize and focus on specific network information. Scalability of the NetFlow analysis is optimized and Flexible NetFlow gives the opportunity to track the important information for the organization. By targeting specific information the amount of information will be reduced and the number of flows being exported reduced, allowing enhanced scalability and aggregation. If for instance the user was interested in TCP application analysis, in Flexible NetFlow the user would configure the tracking of the NetFlow field's *source and destination IP addresses, TCP source and destination ports* and NetFlow will examine the packets for this information. This information will effectively show who is sending and receiving the traffic per application port. In traditional NetFlow, packet information is used to create flows but this information is fixed and not configurable by the user. In traditional NetFlow aggregation comes with the expense of lost information but in Flexible NetFlow the user can actually track multiple sets of information to make sure all flow information in the network is captured efficiently. In the above example the user only needs four NetFlow fields to track application usage and this is contrasted to 7 fields in traditional NetFlow. In traditional NetFlow, the user must track the 7 key fields and each field tracked leads to a greater number of flows.

An Example of Security Detection with Flexible Netflow

Flexible NetFlow is an excellent attack detection tool with capabilities to track all parts of the IPv4 header and even packet sections, and characterize this information into flows. It is expected that security detection systems will listen to NetFlow data and upon finding an issue in the network, create a virtual bucket or virtual cache that will be configured to track specific information and pinpoint details about the attack pattern or worm propagation. The capability to create caches on the fly with specific information combined with input filtering (ie: filtering all flows to a specific destination) allows Flexible NetFlow to be a better security detection tool than current flow technologies. It is expected common attacks such as port scans for worm target discovery and worm propagation will be tracked in Flexible NetFlow. Let's discuss a common simpler attack, in which TCP flags are used to flood open TCP requests to a destination server (ie: SYN flood attack). The attacking device will send a stream TCP SYN's to a given destination address but never send the ACK in response to the servers SYN-ACK as part of the TCP 3-way handshake. The flow information needed for security monitor requires the tracking of three key fields: destination address or subnet, TCP flags and packet count. The security device may be monitoring general NetFlow information and this data may trigger a detailed view of this particular attack. The detailed Flow Monitor might include input filtering to limit what traffic is visible in the NetFlow cache along with the tracking of the specific information to diagnose the TCP based attack. In this case, the user may want to filter all flow information to the server destination address or subnet to limit the amount of information the security server needs to evaluate. If the security detection server decided it understood this attack, it might then program another virtual cache or bucket to export payload information or sections of packets to take a deeper look at a signature within the packet. The above is just one of many possible examples of how Flexible NetFlow can be used to detect security incidents.

What are the Key Components Within Flexible Netflow?

NetFlow has a number of key components:

- NetFlow cache
- NetFlow Flow Record
- Flow export timers
- NetFlow export format
- NetFlow server for collection and reporting

The *NetFlow cache* stores flow information. The *Flow Record* is created by inspecting a packet and a description of packet information is added to the NetFlow cache as the Flow Record. NetFlow exports or pushes flow information to the NetFlow reporting server. This is in contrast to SNMP pull or the polling model to retrieve data. The Flow Records in the cache will expire or terminate and be exported to a NetFlow collector and be used to create management reports. The flows will expire based on timers and if the flow is inactive (no new packets and bytes for the flow) it will be exported. Flow timers are discussed later in the document.

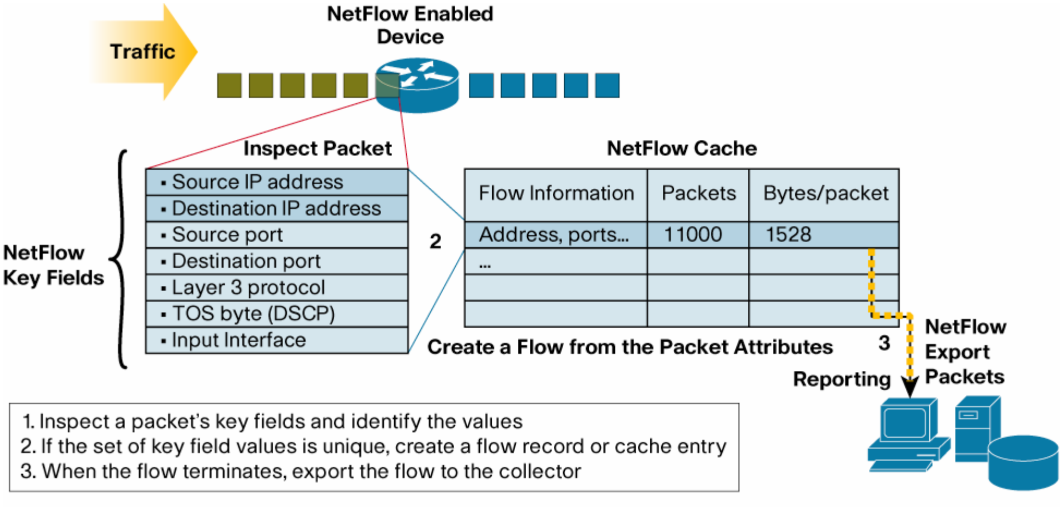
What is a Flexible NetFlow Key Field?

Each packet that is forwarded within a router or switch is examined for a set of IP packet attributes. These attributes are the IP packet identity or *key fields* for the flow and determine if the packet information is unique or similar to other packets.

Traditionally, an IP flow is based on a set of seven IP packet attributes. This set of key fields is tracked and if the set of values for these fields are unique, a new Flow Record is created in the NetFlow cache.

All packets with the same source/destination IP address, source/destination ports, protocol, interface and class of service are grouped into a flow and then packets and bytes tallied. This methodology of flow characterization or determining a flow is scalable because a large amount of network information is condensed into a database of NetFlow information called the NetFlow cache. The NetFlow cache is associated with a flexible Flow Monitor capability that will be discussed in more detail within this document. Figure 2 shows the basic components of NetFlow including the flow key fields, NetFlow cache and reporting server.

Figure 2. Creating a Flow in the NetFlow Cache



Flow Information	Packets	Bytes/packet
Address, ports...	11000	1528
...		

1. Inspect a packet's key fields and identify the values
2. If the set of key field values is unique, create a flow record or cache entry
3. When the flow terminates, export the flow to the collector

What is a Flexible NetFlow Non-Key Field?

Additional information can be added to the Flow Record and this information is named *non-key fields*. Non-key fields are added to the flow entry in the NetFlow cache and exported. The non-key fields are not used to create or characterize the flows but are exported and just added to the flow. In Flexible NetFlow, non-key fields are also configurable by the user. If a field is non-key, normally only the first packet of the flow is used for the value in this field.

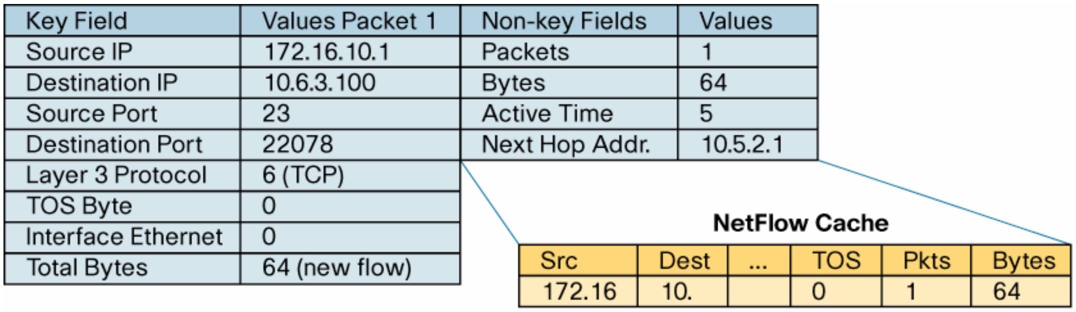
Typical non-key NetFlow fields include:

- Flow timestamps to understand the life of a flow; timestamps are useful for calculating packets and bytes per second
- Next hop IP addresses including BGP routing Autonomous Systems (AS)
- Subnet mask for the source and destination addresses to calculate prefixes
- TCP flags to examine TCP handshakes

Creating a Flow with Key and Non-key Fields

Figure 3 is an example of flow creation based on NetFlow key and non-key fields. The first packet is inspected and the key field values are found and the set of key field values is unique within the flow cache and a flow is created.

Figure 3. Packet 1 has Key and Non-Key Field Values in the Netflow Cache



Key Field	Values Packet 1
Source IP	172.16.10.1
Destination IP	10.6.3.100
Source Port	23
Destination Port	22078
Layer 3 Protocol	6 (TCP)
TOS Byte	0
Interface Ethernet	0
Total Bytes	64 (new flow)

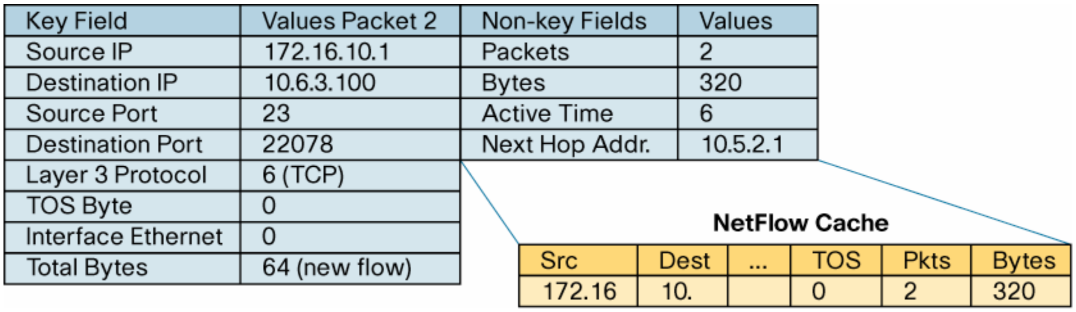
Non-key Fields	Values
Packets	1
Bytes	64
Active Time	5
Next Hop Addr.	10.5.2.1

NetFlow Cache

Src	Dest	...	TOS	Pkts	Bytes
172.16	10.		0	1	64

Packet 2 is then inspected and the values of the key fields are the same as packet 1 and not unique, so packets and bytes are incremented for the existing flow. Figure 4 shows how the packets and bytes are incremented for the flow.

Figure 4. Packet 2 Key and Non-key Field Values and the NetFlow Cache



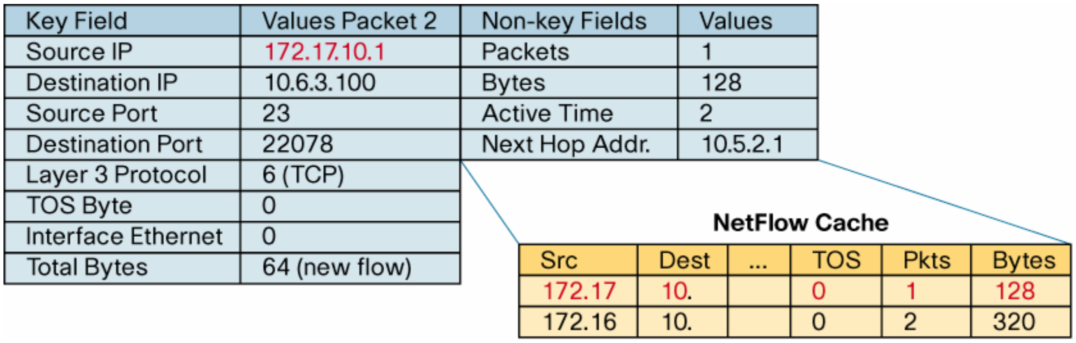
Key Field	Values Packet 2	Non-key Fields	Values
Source IP	172.16.10.1	Packets	2
Destination IP	10.6.3.100	Bytes	320
Source Port	23	Active Time	6
Destination Port	22078	Next Hop Addr.	10.5.2.1
Layer 3 Protocol	6 (TCP)		
TOS Byte	0		
Interface Ethernet	0		
Total Bytes	64 (new flow)		

NetFlow Cache

Src	Dest	...	TOS	Pkts	Bytes
172.16	10.		0	2	320

Packet 3 is then inspected and the value of source address key field changes, and when the flow cache is inspected, the set of key field values is unique and therefore a new flow is created in the NetFlow cache. Figure 5 shows information for packet 3.

Figure 5. Packet 3 Key and Non-key Field Values and the NetFlow Cache



Key Field	Values Packet 2	Non-key Fields	Values
Source IP	172.17.10.1	Packets	1
Destination IP	10.6.3.100	Bytes	128
Source Port	23	Active Time	2
Destination Port	22078	Next Hop Addr.	10.5.2.1
Layer 3 Protocol	6 (TCP)		
TOS Byte	0		
Interface Ethernet	0		
Total Bytes	64 (new flow)		

NetFlow Cache

Src	Dest	...	TOS	Pkts	Bytes
172.17	10.		0	1	128
172.16	10.		0	2	320

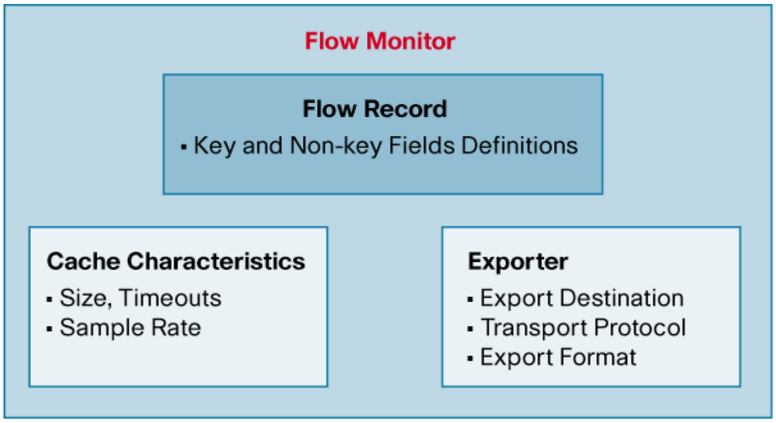
The set of key field values are used to determine if a flow is unique and should be tracked as a new flow or if packets and bytes should be tallied for an existing flow.

Flexible NetFlow will allow the user to select what key and non-key fields to define flows. This capability allows the user flexibility, aggregation and scalability beyond traditional NetFlow.

What is a Flexible NetFlow Flow Monitor?

A Flexible NetFlow Flow Monitor describes the NetFlow cache or information stored in the cache. The Flow Monitor contains the Flow Records or key and non-key fields within the cache. Also, part of the Flow Monitor is the Flow Exporter which contains information about the export of NetFlow information including the destination address of the NetFlow collector. The Flow Monitor includes various cache characteristics including the timers for exporting, the size of the cache and if required, the packet sampling rate. Figure 6 depicts the Flow Monitor and its components.

Figure 6. Flow Monitor Components



What is a Flexible NetFlow Flow Record?

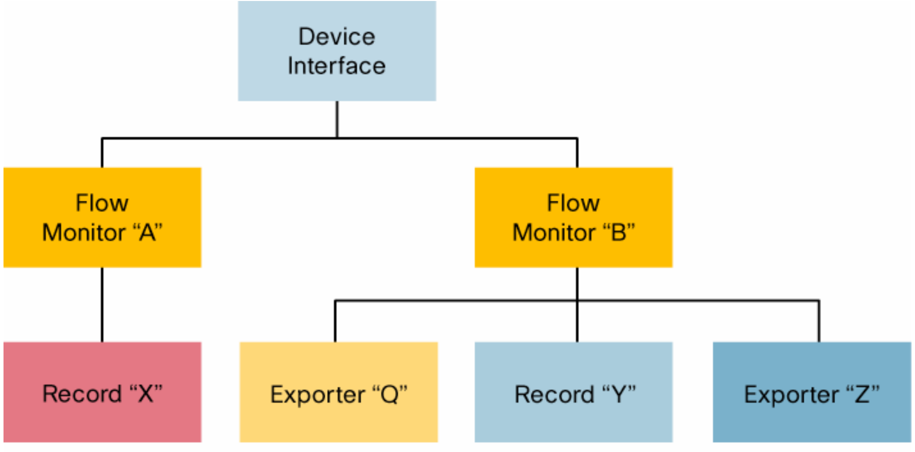
A Flow Record is a set of key and non-key NetFlow field values used to characterize flows in the NetFlow cache. Flow Records may be pre-defined for ease of use or customized and user defined. A typical pre-defined record will aggregate flow data and allow users to target common applications for NetFlow. User defined records will allow selection of specific key or non-key fields in the Flow Record. The user defined field is the key to Flexible NetFlow allowing a wide range of information to be characterized and exported by NetFlow. It is expected that different network management applications will support specific user defined and pre-defined Flow Records based on what they are monitoring (ie: security detection, traffic analysis, capacity planning).

What is a Flexible NetFlow Exporter?

There are two primary methods to access NetFlow data: the Command Line Interface (CLI) with show commands or utilizing an application reporting tool receiving export. NetFlow export, unlike SNMP, polling pushes information periodically to the NetFlow reporting collector. In general, the NetFlow cache is constantly filling with flows, and software in the router or switch is searching the cache for flows that have terminated or expired, and these flows are exported to the NetFlow collector server. Flow export is optional but it’s the only method to get a complete view of all NetFlow data in the Cisco device. The Flexible NetFlow Exporter allows the user to define where the export can be sent, the type of transport for the export and properties for the export. Multiple exporters can be configured per Flow Monitor or the same exporter can be used by multiple monitors. For example, the user may want to send the same data to a billing and also a traffic analysis server. The number of exporters is only limited by the resources on the network device. The Flexible NetFlow Exporter also has the capability to send optional data such as tables of information to a NetFlow collector. An example of table export would be the interface If-index to interface name mapping. The exporter can also allow the class of service to be marked for the export stream. The exporter will support various export formats including v5, v9 and the IETF IP Flow Information Export standard (IPFIX). The exporter can support various transport protocols including UDP, Stream Control Transmission Protocol (SCTP).

Figure 7 shows the flexibility of Flexible NetFlow including the ability to create different Flow Monitors per device interface and the ability to have different Flow Records and exporters per Flow Monitor.

Figure 7. Flexible NetFlow Flow Monitor, Export and Record Definition



What are the Netflow Cache Characteristics and Flow Timers?

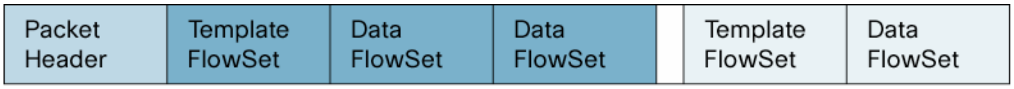
Flows are continuously being created, are tracked and then expire and are exported from the NetFlow cache to a reporting server. A flow is ready for export when it is inactive for a certain time (ie: no new packets received for the flow); or if the flow is long lived (active) and lasts greater than the active timer (ie: long FTP download). Their are timers to determine if a flow is inactive or if a flow is long lived, and the default for the inactive flow timer is 15 seconds and the active flow timer is 30 minutes. All the timers for export are configurable.

Each cache can also be configured for packet sampling and therefore a subset of packets can be randomly sampled from the traffic stream and used to characterize the network traffic. Packet sampling is effective for capacity planning, where an approximate view of network traffic may be enough to give a good indication of traffic capacity. Packet sampling can be used for high speed interfaces (ie: OC-192) where creating flows for every packet is not possible. Packet sampling can be an effective method to decrease the volume of NetFlow export and decrease CPU utilization. Many Cisco platforms can support full NetFlow (non-sampled) but in some environments sampling is a necessity.

What is the NetFlow Version 9 Export Format?

There are various formats for the export packet and these are commonly called the *export version*. The export versions are well documented formats including version 5, 7, and 9. The most common format used is NetFlow export version 5, but version 9 is the latest Cisco invented format and has some advantages for key technologies such as security, traffic analysis and multicast. *Without version 9 export format, Flexible NetFlow would not be possible.* NetFlow version 5 is a fixed export format which means the data in export version 5 cannot be expanded beyond what is available today. The key to Flexible NetFlow is the ability to allow flexibility in the configuration of the data gathered and monitored by NetFlow. The best method to export a wide range of information from the packet is to use NetFlow version 9, which is a generic format to export any information from a network device. NetFlow version 9 uses a template to describe the data it will export and the data is specifically associated with the template. The NetFlow Version 9 record format consists of a packet header followed by at least one or more template or data FlowSets. A template FlowSet provides a description of the fields that will be present in future data FlowSets . These data FlowSets may occur later within the same export packet or in subsequent export packets. Template and data FlowSets can be intermingled within a single export packet, as illustrated in Figure 8.

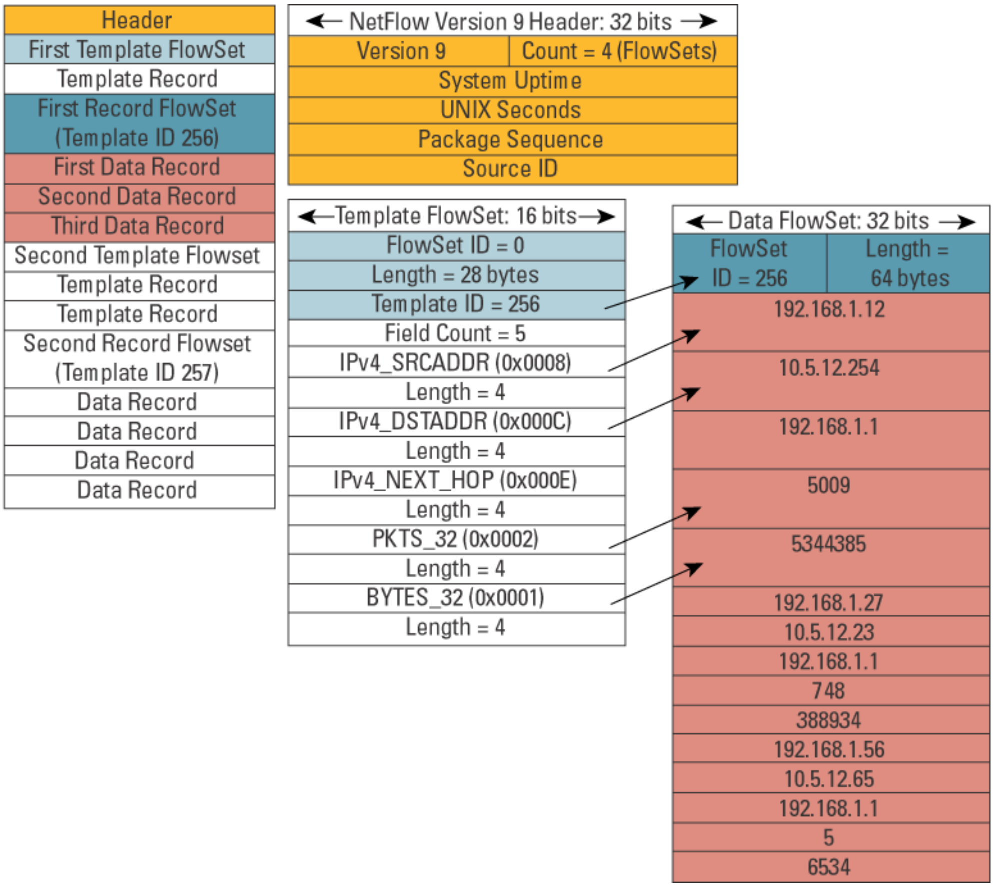
Figure 8. NetFlow Version 9 Export Packet



NetFlow version 9 will periodically export the template data, so the collector will understand what data is to be sent and also export the data flow set for the template. The key advantage to Flexible NetFlow is the user configures a Flow Record which is effectively converted to a version 9 template and then forwarded to the collector.

Figure 9 is a detailed example of the NetFlow export format including the header, template and data FlowSets.

Figure 9. Detailed View of a NetFlow Version 9 Export Format



How to Configure Flexible NetFlow Exporter, Record and Flow Monitor?

Configuring Flexible NetFlow can be quite easy and the user can use pre-defined or user defined Flow Records. The first example will demonstrate pre-defined Flow Records and the configuration of traditional NetFlow export using NetFlow version 9. By default, the export version used in Flexible NetFlow is NetFlow version 9.

The key components that need to be configured:

1. Configure the exporter if it is to export to a collector.
2. Configure the Flow Monitor with the pre-defined Flow Record and Flow Exporter attached to the monitor.

3. Add the Flow Monitor to the interface to monitor either ingress (input) or egress (output traffic).

   In example one, the Flow Exporter is named “export-to-server” and is attached under the Flow Monitor “my-flow-monitor”. Also notice the pre-defined Flow Record chosen is “original-netflow”. This record effectively configures Flexible NetFlow to behave like traditional NetFlow which provides backward compatibility.

Example 1: Predefined Flow Record Configuration

```
flow exporter export-to-server
 destination 172.16.1.1
flow monitor my-flow-monitor
 record netflow-orginal
 exporter export-to-server
interface Ethernet 1/0
 ip flow monitor my-flow-monitor input
```

Keep in mind all pre-defined IPv4 and IPv6 flow records are available for the user to configure:

```
Router(config-flow-monitor)# record netflow ipv4 ?
  as                       AS aggregation schemes
  as-tos                   AS and TOS aggregation schemes
  bgp-nexthop-tos          BGP next-hop and TOS aggregation schemes
  destination-prefix       Destination Prefix aggregation schemes
  destination-prefix-tos   Destination Prefix and TOS aggregation schemes
  original-input           Traditional IPv4 input NetFlow
  original-output          Traditional IPv4 output NetFlow
  prefix                   Source and Destination Prefixes aggregation
                           schemes
  prefix-port              Prefixes and Ports aggregation scheme
  prefix-tos               Prefixes and TOS aggregation schemes
  protocol-port            Protocol and Ports aggregation scheme
  protocol-port-tos        Protocol, Ports and TOS aggregation scheme
  source-prefix            Source AS and Prefix aggregation schemes
  source-prefix-tos        Source Prefix and TOS aggregation schemes
```

In the next example, a user defined Flow Record will be created and customized with the information tracked by Flexible NetFlow. A good example would be the idea outlined earlier in the white paper to monitor how much traffic that will be used per TCP application. In contrast to traditional NetFlow, in this case Flexible NetFlow will monitor just the key fields or information desired, which increases scalability by reducing the number key fields that are necessary for flow creation. In this example, the user also wants to track packets and bytes. The “match” keyword is used to denote the field as a key field and as discussed above, is used to characterize and create flows. The “collect” keyword is used to denote the non-key field and will be used for information that is to be added to the flow, but this information is not used when creating the flow, but exported along with the flow.

The key components that need to be configured:

1. Configure the user defined Flow Record with key and non-key fields
2. Configure the exporter if it is to export to a collector
3. Configure the Flow Monitor with the user defined Flow Record and Flow Exporter attached to the monitor
4. Add the Flow Monitor to the interface to monitor either ingress (input) or egress (output traffic)

Example 2 shows how to configure a user defined Flow Record with the Flow Record created named “app-traffic-analysis” with the Flow Exporter and Flow Monitor named the same as in example 1.

Example 2: User Defined Flow Record Configuration

```
flow record app-traffic-analysis
 description This flow record tracks application usage
 match transport tcp destination-port
 match transport tcp source-port
 match ipv4 destination address
 match ipv4 source address
 collect counter bytes
 collect counter packets
flow exporter export-to-server
 destination 172.16.1.1
flow monitor my-flow-monitor
 record app-traffic-analysis
 exporter export-to-server
interface Ethernet 1/0
 ip flow monitor my-flow-monitor input
```

Another nice feature of Flexible NetFlow is once a user defined Flow Record is created, this record will become available as a pre-defined record that can be easily selected for future use.

What Packet Fields can be Tracked in Flexible NetFlow?

Flexible NetFlow is designed to track information from layer 2 to layer 7 and extract this information from the IP packet and create flows. The following tables outline the key and non-key information that is available in the first release of Flexible NetFlow. There are a large number of fields that can be tracked and different types of applications will utilize a portion of the information available. This information includes routing, transport, and IPv4 packet information. Tables 2 though 6 provide information that can be tracked within Flexible NetFlow.

Table 2. IPv4 Information that is Tracked with Flexible NetFlow

Field	Summary
Destination Address	Configure IPv4 destination address fields as a key or non-key field.
Source Address	Configures the IPv4 source address as a key or non-key field.
IP Destination Mask	Configures the IPv4 destination address mask for the IPv4 destination address as a key or non-key field.
IP Source Mask	Configures the IPv4 source address mask for the IPv4 destination address as a key or non-key field.
Minimum-mask source	Configures a mask for the minimum-mask keyword.
Dscp	Configures the IPv4 DSCP (part of ToS) as a key or non-key field.
Minimum-mask destination	Configures a mask for the minimum-mask keyword
Prefix Destination	Configures the IPv4 address prefix for the IPv4 destination address as a key or non-key field.
Prefix Source	Configures the IPv4 address prefix for the IPv4 source address as a key or non-key field.
Fragmentation flags	Configures the IPv4 fragmentation flags as a key or non-key field.
Fragmentation offset	Configures the IPv4 fragmentation offset as a key or non-key field.
Header-length	Configures the IPv4 header length (in 32 bit words) as a key or non-key field.
Id	Configures the IPv4 ID as a key or non-key field.
Options	Configures the Bitmap representing which IPv4 options have been seen as a key or non-key field.
Precedence	Configures the IPv4 precedence (part of ToS) as a key or non-key field.
Protocol	Configures the IPv4 protocol as a key or non-key field.
Section	Configures NetFlow to export up to 1200 bytes of the IP packet. A packet section can start at the IP payload or header.
Header size	Configures the number of bytes of raw data starting at the IPv4 header to use as a key or non-key field.
Payload size	Configures the number of bytes of raw data starting at the IPv4 payload to use as a key or non-key field.
ToS	Configures the IPv4 type of service as a key or non-key field.
Total-length	Configures the IPv4 total length as a key or non-key field.
TTL	Configures the IPv4 time to live as a key or non-key field.
Version	Configures the IP version from IPv4 header as a key or non-key field.

Table 3. IPv4 Routing Fields that are Available in Flexible NetFlow

Field	Summary
Destination	Configures one or more of the destination routing attributes fields as a key or non-key field. Such as destination AS or BGP PA traffic index.
AS	Configures the destination AS field as a key or non-key field. Note: This goes with the [match | collect] routing destination as command.
Peer	Configures the destination AS number of the peer network as a key or non-key field.
Traffic-index	Configures the BGP destination traffic index as a key or non-key field.
Forwarding-status	Configures the forwarding status as a key or non-key field. Was the packet forwarded or dropped for the flow.
is-multicast	Configures if the traffic is multicast as a key or non-key field.
Next-hop address ipv4	Configures the next hop value as a key or non-key field.
BGP	Configures if the IPv4 address of the next hop is a BGP destination as a key or non-key field.
Source	Configures one or more of the source routing attributes fields as a key or non-key field. Such as source AS or BGP PA traffic index.
BGP AS	Configures the source AS field as a key or non-key field

Table 4. IPv4 Transport Fields that are Available in Flexible NetFlow

Field	Summary
Destination-port	Configures the transport destination port as a key or non-key field.
icmp-ipv4 code	Configures the IPv4 ICMP code as a key or non-key field.
icmp-ipv4 type	Configures the IPv4 ICMP type as a key or non-key field.
igmp type	Configures timestamps based on the sys-uptime as a key or non-key field.
source-port	Configures the Transport source port as a key or non-key field.
tcp	Specify one or more of the TCP fields as a key or non-key field.
acknowledgement-number	Configures the TCP acknowledgement number as a key or non-key field.
destination-port	Configures the TCP destination port as a key or non-key field.
flags	Specify one or more of the TCP flag fields as a key or non-key field.
ack	Configures the TCP acknowledgement flag as a key or non-key field.
cwr	Configures the TCP congestion window reduced flag as a key or non-key.
ece	Configures the TCP ECN echo flag as a key or non-key field.
fin	Configures the TCP finish flag as a key or non-key field.
psh	Configures the TCP push flag as a key or non-key field.
rst	Configures the TCP reset flag as a key or non-key field.
syn	Configures the TCP synchronize flag as a key or non-key field.
Urg	Configures the TCP urgent flag as a key or non-key field.
header-length	Configures the TCP header length (in 32 bit words) as a key or non-key field.
sequence-number	Configures the TCP sequence number as a key or non-key field.
source-port	Configures the TCP source port as a key or non-key field.
urgent-pointer	Configures the TCP urgent pointer as a key or non-key field.
window-size	Configures the TCP window size as a key or non-key field.
udp	Specify one or more of the UDP fields as a key or non-key field.
destination-port	Configures the UDP destination port as a key or non-key field.
message-length	Configures the UDP message length as a key or non-key field.
source-port	Configures the UDP source port as a key or non-key field.

Table 5. IPv6 Information That is Tracked with Flexible NetFlow

Field	Summary
Destination Address	Configure IPv6 destination address fields as a key or non-key field.
Source Address	Configures the IPv6 source address as a key or non-key field.
IP Destination Mask	Configures the IPv6 destination address mask for the IPv6 destination address as a key or non-key field.
IP Source Mask	Configures the IPv6 source address mask for the IPv6 destination address as a key or non-key field.
Minimum-mask source	Configures a mask for the minimum-mask keyword.
Dscp	Configures the IPv6 DSCP (part of TOS) as a key or non-key field.
Traffic Class	Configures the IPv6 Traffic Class as a key or non-key field.
Flow Label	Configures the IPv6 Flow Label as a key or non-key field.
Header-length	Configures the IPv6 header length (in 32 bit words) as a key or non-key field.
Payload-length	Configures the IPv6 payload length (in 32 bit words) as a key or non-key field.
Hop-Limit	Configures the IPv6 Hop-Limit as a key or non-key field.
Option Header	Configures the Bitmap representing which IPv6 options have been seen as a key or non-key field.
Protocol	Configures the IPv6 protocol as a key or non-key field.

Field	Summary
Section	Configures NetFlow to export up to 1200 bytes of the IP packet. A packet section can start at the IP payload or header.
Header size	Configures the number of bytes of raw data starting at the IPv6 header to use as a key or non-key field.
Payload size	Configures the number of bytes of raw data starting at the IPv6 payload touse as a key or non-key field.
Tos	Configures the IPv6 type of service as a key or non-key field.
Total-length	Configures the IPv6 total length as a key or non-key field.
Next-Header	Configure the IPv6 next-header as a key or non-key field
Version	Configures the IP version from IPv6 header as a key or non-key field.

Table 6. IPv6 Routing Fields that are Available in Flexible NetFlow

Field	Summary
Destination	Configures one or more of the destination routing attributes fields as a key or non-key field. Such as destination AS or BGP PA traffic index
AS	Configures the destination AS field as a key or non-key field. Note This goes with the [match | collect] routing destination as command.
Peer	Configures the destination AS number of the peer network as a key or non-key field.
Traffic-index	Configures the BGP destination traffic index as a key or non-key field.
Forwarding-status	Configures the forwarding status as a key or non-key field. Was the packet forwarded or dropped for the flow.
is-multicast	Configures if the traffic is multicast as a key or non-key field.
Next-hop address ipv4	Configures the next hop value as a key or non-key field.
BGP	Configures if the IPv6 address of the next hop is a BGP destination as a key or non-key field.
Source	Configures one or more of the source routing attributes fields as a key or non-key field. Such as source AS or BGP PA traffic index
AS	Configures the source AS field as a key or non-key field. peer field.

Table 7. IPv6 Transport Fields that are Available in Flexible NetFlow

Field	Summary
Destination-port	Configures the transport destination port as a key or non-key field.
icmp-ipv6 code	Configures the IPv6 ICMP code as a key or non-key field.
icmp-ipv6 type	Configures the IPv6 ICMP type as a key or non-key field.
source-port	Configures the Transport source port as a key or non-key field.
tcp	Specify one or more of the TCP fields as a key or non-key field.
acknowledgement-number	Configures the TCP acknowledgement number as a key or non-key field.
destination-port	Configures the TCP destination port as a key or non-key field.
flags	Specify one or more of the TCP flag fields as a key or non-key field.
ack	Configures the TCP acknowledgement flag as a key or non-key field.
cwr	Configures the TCP congestion window reduced flag as a key or non-key field.
ece	Configures the TCP ECN echo flag as a key or non-key field.
fin	Configures the TCP finish flag as a key or non-key field.
psh	Configures the TCP push flag as a key or non-key field.
rst	Configures the TCP reset flag as a key or non-key field.
syn	Configures the TCP synchronize flag as a key or non-key field.
Urg	Configures the TCP urgent flag as a key or non-key field.
header-length	Configures the TCP header length (in 32 bit words) as a key or non-key field.
sequence-number	Configures the TCP sequence number as a key or non-key field.

Field	Summary
source-port	Configures the TCP source port as a key or non-key field.
urgent-pointer	Configures the TCP urgent pointer as a key or non-key field.
window-size	Configures the TCP window size as a key or non-key field.
udp	Specify one or more of the UDP fields as a key or non-key field.
destination-port	Configures the UDP destination port as a key or non-key field.
message-length	Configures the UDP message length as a key or non-key field.
source-port	Configures the UDP source port as a key or non-key field.

Table 8. Other Fields that can be Tracked in Flexible NetFlow

Field	Summary
direction	Configures the direction the flow was monitored in as a key or non-key field.
sampler	Configures the sampler ID as a key or non-key field.
input	Configures the direction the input interface as a key or non-key field.
output	Configures the direction the output interface as a key or non-key field.
sys-uptime	Configures timestamps based on the sys-uptime for the beginning and end of a flow as non-key fields
Packets	Configures to count the number of packets as a non-key fields
Bytes	Configures to count the number of bytes as a non-key fields

What Show Commands Can be Used to Track Flexible NetFlow?

The following table 6 is a list of show commands available within Flexible NetFlow. The user can view information including records, interface, exporters and Flow Monitors and flows in the cache.

Table 9. Show Commands Available within Flexible NetFlow

Show Command	Usage
Show run flow [exporter | monitor | record]	Parses the show run command for output
Show flow [exporter | interface | monitor | record]	Shows detailed information about the Flexible NetFlow component
Show flow monitor [*name of monitor*] cache	Shows the contents of the Flexible NetFlow cache in comma separated format (CSV), table or record (list) format.

The following is an example show command to view the configured Flow Monitor, record or exporter by parsing the running configuration. The advantage of this command is the output can be cut and pasted directly into the device in configuration mode.

```
R3#show run flow ?
 exporter Show Flow Exporter configuration
 monitor Show Flow Monitor configuration
 record Show Flow Record configuration
```

This command will parse the running config and show the details of a Flow Monitor:

```
R3#show run flow monitor My-flow-monitor
Building configuration...
Current configuration:
!
flow monitor My-flow-monitor
```

```
 record app-traffic-analysis
 exporter export-to-server
!
End
```

This command will parse the running config and show the details of a Flow Record:

```
R3#show run flow record app-traffic-analysis
Building configuration...
Current configuration:
!
flow record app-traffic-analysis
 description This flow record tracks application useage
 match transport tcp destination-port
 match transport tcp source-port
 match ipv4 destination address
 match ipv4 source address
 collect counter packets
!
end
```

This command will parse the running config and show the details of the Flow Exporter:

```
R3#show run flow exporter
Building configuration...
Current configuration:
!
flow exporter export-to-server
 destination 172.16.1.1
!
end
```

Another method to view a Flow Record, exporter, interface or Flow Monitor is using the "show flow" command and this provides details beyond just paring the configuration.

```
R3# show flow ?
 exporter Flow Exporter information
 interface Flow interface information
 monitor Flow Monitor information
 record Show Flow Record configuration
```

The following command shows the details of the Flow Monitor named my-flow-monitor. It includes information on the size of the cache and flow export timers associated with the Flow Monitor.

```
R3#show flow monitor my-flow-monitor
```

```
Flow Monitor my-flow-monitor:
 Description: User defined
 Flow Record: app-traffic-analysis
 Flow Exporter: export-to-server
 Cache:
 Type: normal
 Status: allocated
 Size: 4096 entries / 196620 bytes
 Inactive Timeout: 15 secs
 Active Timeout: 1800 secs
 Update Timeout: 1800 secs
```

The following command shows the details of the Flow Record including fields that are tracked. It is also possible to see details of the NetFlow version 9 export templates if that level of detail is required.

```
R3#show flow record app-traffic-analysis
flow record app-traffic-analysis:
 Description: This flow record tracks application usage
 No. of users: 1
 Total field space: 28 bytes
 Fields:
 match ipv4 source address
 match ipv4 destination address
 match transport source-port
 match transport destination-port
 collect counter packets
 collect counter bytes
```

The details of the exporter are also available including the transport protocol, port used, COS service setting and time to live for the export packet.

```
R3#show flow exporter export-to-server
Flow Exporter export-to-server:
 Description: User defined
 Tranport Configuration:
 Destination IP address: 172.16.1.1
 Source IP address: 172.16.6.2
 Transport Protocol: UDP
 Destination Port: 9995
 Source Port: 49750
 DSCP: 0x0
 TTL: 255
```

In Flexible NetFlow new show commands have been developed to show the details of the NetFlow cache. These commands replace the old “show ip cache flow” command traditionally used in NetFlow. The cache can be shown in a number of formats including record or list format, table or common separated format. The following are examples of the output formats.

The following example shows the cache for the Flow Monitor named my-flow-monitor:

```
R3#show flow monitor my-flow-monitor cache table
 Cache type: Normal
 Cache size: 4096
 Current entries: 5
 High Watermark: 6
 Flows added: 94
 Flows aged: 89
 - Active timeout ( 1800 secs) 0
 - Inactive timeout ( 15 secs) 89
 - Event aged 0
 - Watermark aged 0
 - Emergency aged 0
IPV4 SRC ADDR IPV4 DST ADDR TRNS SRC PORT TRNS DST PORT bytes pkts time
first time last
=============== =============== ============= ============= ====== ======
========== ==========
50.0.0.4 172.16.1.1 20 20 348440 8711 1375212 1418236
50.0.0.5 172.16.1.1 20 20 174360 4359 1375232 1418232
172.16.10.2 172.16.1.1 0 771 4704 84 1375244 1418484
50.0.0.5 50.0.0.3 27577 179 59 1 1410144 1410144
50.0.0.4 50.0.0.3 179 13134 59 1 1412136 1412136
```

The following shows the NetFlow cache details in the record or list format. This is the default format to view the cache.

```
R3#show flow monitor my-flow-monitor cache record
 Cache type: Normal
 Cache size: 4096
 Current entries: 4
 High Watermark: 6
 Flows added: 95
 Flows aged: 91
 - Active timeout ( 1800 secs) 0
 - Inactive timeout ( 15 secs) 91
 - Event aged 0
 - Watermark aged 0
 - Emergency aged 0
IPV4 SOURCE ADDRESS: 50.0.0.4
IPV4 DESTINATION ADDRESS: 172.16.1.1
TRNS SOURCE PORT: 20
TRNS DESTINATION PORT: 20
counter bytes: 531960
counter packets: 13299
timestamp first: 1375212
timestamp last: 1441264
IPV4 SOURCE ADDRESS: 50.0.0.5
IPV4 DESTINATION ADDRESS: 172.16.1.1
TRNS SOURCE PORT: 20
TRNS DESTINATION PORT: 20
counter bytes: 266160
counter packets: 6654
timestamp first: 1375232
timestamp last: 1441248
```

The following is a view of the NetFlow cache in comma separated format which can be imported directly into a database or spreadsheet.

```
R3#show flow monitor my-flow-monitor cache csv
 Cache type: Normal
 Cache size: 4096
 Current entries: 4
 High Watermark: 6
 Flows added: 96
 Flows aged: 92
 - Active timeout ( 1800 secs) 0
 - Inactive timeout ( 15 secs) 92
 - Event aged 0
 - Watermark aged 0
 - Emergency aged 0
IPV4 SRC ADDR,IPV4 DST ADDR,TRNS SRC PORT,TRNS DST PORT,bytes,pkts,time first,time last
50.0.0.4,172.16.1.1,20,20,659520,16488,1375212,1457256
```

```
50.0.0.5,172.16.1.1,20,20,330080,8252,1375232,1457256
172.16.10.2,172.16.1.1,0,771,8960,160,1375244,1457632
172.16.7.2,224.0.0.9,520,520,92,1,1452444,1452444
```

What Types of Flexible NetFlow Caches are Available?

There are three types of flow caches that can be used in Flexible NetFlow:

- Normal cache
- Permanent caches
- Immediate cache

In traditional NetFlow, the only cache that is available is what is called the normal cache in Flexible NetFlow. The normal cache uses flow timers to expire and export flows to a NetFlow collector. The normal cache in Flexible NetFlow does have an advantage over traditional NetFlow. The active timer for the cache (ie: the timer that tracks long flows) can be set as low as 1 second and in traditional NetFlow the minimum value was 60 seconds. This is a great advantage for tracking security incidents where open or partial flows might be recorded (ie: SYN flood attack). Additionally in Flexible NetFlow, two other cache types are now available—the permanent cache and immediate cache.

The permanent cache is very different from a normal cache and will be most useful for accounting or security monitoring. The permanent cache will be a fixed size chosen by the user. After the permanent cache is full, all new flows will be dropped. A good example of the usage of this type of cache would be the tracking of traffic matrix, where a fixed set of flows are used to track traffic between specific end points. In this case, the cache would establish these flows and continue to track them until the counters reached size limits. This is similar to access-list or interface counters. The permanent cache will be configured to periodically export its information to the collector reporting server.

The immediate export cache will be very effective for security monitoring or when the user wants each packet to effectively create a new flow. A good example of an immediate cache is when using packet section export in Flexible NetFlow. Flexible NetFlow can export sections of a packet in version 9 format. The section of packet exported will be described in the NetFlow version 9 templates (ie: 1000 bytes from IP payload) and then the sections of packet to follow in the data sets within version 9.

The following is an example of packet section export and the configuration of an immediate cache type in Flexible NetFlow. The example might be related to a UDP port 53 anomaly spike in traffic to a DNS. The user is going to use a deep packet inspection to see if the queries are legitimate. The Flow Monitor is tracking source and destination IP addresses along with UDP destination ports and this information will be exported along with the packet section. Since this is an immediate cache, each packet will effectively create a flow. The Flow Exporter is configured to allow class of service based export with the dscp for the export stream set to af43. The cache is configured for a maximum size of 1000 flows and the time to live for the export stream is set to 20 hops.

```
flow record packet-section
 match ipv4 section payload size 900
 match transport udp destination-port
 match ipv4 destination address
 match ipv4 source address
 collect counter packets
!
flow exporter my-exporter
 destination 1.1.1.1
 dscp af43
flow monitor immediate-export
 record packet-section
 exporter my-exporter
 cache type immediate
 cache entries 1000
 ttl 20
```

Summary

Flexible NetFlow is an important technology available in Cisco devices to help with visibility into how network assets are being used and the network behavior. Flexible NetFlow is an improved NetFlow bringing better scalability, aggregation of data and user customization. Flexible NetFlow will enhance the ability to detect security incidents and understand the behavior of traffic in the network beyond what is possible in other flow based technologies.

For more information on NetFlow visit http://www.cisco.com/go/netflow

For more details on Flexible NetFlow please see the Cisco IOS Software documentation available at: http://www.cisco.com/en/US/products/ps6601/prod_white_papers_list.html

For an overview of basic traditional NetFlow see the “Understanding NetFlow—Technical Overview” document: http://www.cisco.com/en/US/products/ps6601/products_white_paper0900aecd80406232.shtml

For a detailed discussion of traditional NetFlow the services and solutions guide is available: http://www.cisco.com/en/US/products/sw/netmgtsw/ps1964/products_implementation_design_guide09186a00800d6a11.html

CISCO

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

Printed in USA C11-495156-00 09/08