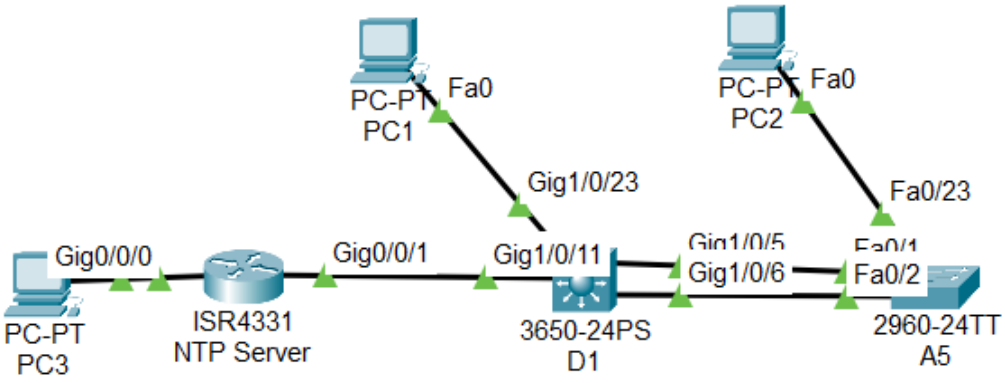
Lab - Implement Flexible Netflow

Topology



Addressing Table

Device	Interface	IP Address	IPv6 Address	IPv6 Link Local
R1	G0/0/1	192.168.1.1/24	2001:db8:acad:1000::1/64	fe80::1:1
	G0/0/0	10.0.0.1/24	2001:db8:acad:10::1/64	fe80::1:2
D1	VLAN 1	192.168.1.2/24	2001:db8:acad:1000::2/64	fe80::d1:1
A1	VLAN 1	192.168.1.3/24		
PC1	NIC	192.168.1.50/24	2001:db8:acad:1000::50/64	EUI-64
PC2	NIC	192.168.1.75/24	Assigned by SLAAC	EUI-64
PC3	NIC	10.0.0.10/24	2001:db8:acad:10::10/64	EUI-64

Objectives

Part 1: Build the Network and Configure Basic Device Settings and Interface Addressing

Part 2: Configure and Verify Flexible Netflow

Part 3: (Optional) Configure and Verify Netflow

Background / Scenario

NetFlow is a Cisco IOS technology that provides statistics on packets flowing through the router. NetFlow is the standard for acquiring IP operational data from IP networks. NetFlow provides data to enable network and security monitoring, network planning, traffic analysis, and IP accounting.

Flexible NetFlow improves on original NetFlow by adding the capability to customize the traffic analysis parameters for your specific requirements. Flexible NetFlow facilitates the creation of more complex configurations for traffic analysis and data export through the use of reusable configuration components.

Note: This lab is an exercise in configuring options available for Flexible Netflow and does not necessarily reflect network troubleshooting best practices.

Note: The routers used with CCNP hands-on labs are Cisco 4221 with Cisco IOS XE Release 16.9.4 (universalk9 image). The switches used in the labs are Cisco Catalyst 3650s with Cisco IOS XE Release 16.9.4 (universalk9 image) and Cisco Catalyst 2960s with Cisco IOS Release 15.2(2) (lanbasek9 image). Other routers, switches, and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and the output produced might vary from what is shown in the labs. Refer to the Router Interface Summary Table at the end of the lab for the correct interface identifiers.

Note: IOS XE does not support classic Netflow. If your lab has ISR G2 series routers, skip Part 2 of this lab and do Part 3, which covers classic Netflow.

Note: Make sure that the switches have been erased and have no startup configurations. If you are unsure, contact your instructor.

Note: The default Switch Database Manager (SDM) template on a Catalyst 2960 does not support IPv6. You must change the default SDM template to the dual-ipv4-and-ipv6 default template using the **sdm prefer dual-ipv4-and-ipv6 default** global configuration command. Changing the template will require a reboot.

Required Resources

- 1 Router (Cisco 4221 with Cisco IOS XE Release 16.9.4 universal image or comparable)
- 1 Switch (Cisco 3650 with Cisco IOS XE Release 16.9.4 universal image or comparable)
- 1 Switch (Cisco 2960 with Cisco IOS Release 15.2(2) lanbasek9 image or comparable)
- 2 PCs (Choice of operating system with terminal emulation program and a packet capture utility installed, such as Wireshark)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet cables as shown in the topology

Instructions

Part 1: Build the Network and Configure Basic Device Settings and Interface Addressing

In Part 1, you will set up the network topology and configure basic settings and interface addressing on routers.

Step 1: Cable the network as shown in the topology.

Attach the devices as shown in the topology diagram, and cable as necessary.

Step 2: Configure basic settings for each device.

a. Console into each device, enter global configuration mode, and apply the basic settings. The startup configurations for each device are provided below.

Router R1

```
hostname R1
no ip domain lookup
ipv6 unicast-routing
banner motd # R1, Implement Flexible Netflow #
line con 0
 exec-timeout 0 0
```

```
 logging synchronous
 exit
line vty 0 4
 privilege level 15
 exec-timeout 0 0
 password cisco123
 login
 exit
interface g0/0/0
 ip address 10.0.0.1 255.255.255.0
 ipv6 address fe80::1:2 link-local
 ipv6 address 2001:db8:acadd:10::1/64
 no shutdown
 exit
interface g0/0/1
 ip address 192.168.1.1 255.255.255.0
 ipv6 address fe80::1:1 link-local
 ipv6 address 2001:db8:acad:1000::1/64
 no shutdown
 exit
ntp master 3
end
```

Switch D1

```
hostname D1
no ip domain lookup
ipv6 unicast-routing
banner motd # D1, Implement Flexible Netflow #
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
line vty 0 4
 privilege level 15
 exec-timeout 0 0
 password cisco123
 login
 exit
interface vlan 1
 ip address 192.168.1.2 255.255.255.0
 ipv6 address fe80::d1:1 link-local
 ipv6 address 2001:db8:acad:1000::2/64
 no shutdown
 exit
ip default-gateway 192.168.1.1
interface g1/0/23
```

```
 spanning-tree portfast
 switchport mode access
 no shutdown
 exit
interface g1/0/11
 spanning-tree portfast
 switchport mode access
 no shutdown
 exit
interface range g1/0/5-6
 switchport trunk encapsulation dot1q
 switchport mode trunk
 channel-group 1 mode active
 no shutdown
 exit
ntp server 192.168.1.1
end
```

Switch A1

```
hostname A1
no ip domain lookup
banner motd # A1, Implement Flexible Netflow #
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
line vty 0 4
 privilege level 15
 exec-timeout 0 0
 password cisco123
 login
 exit
interface vlan 1
 ip address 192.168.1.3 255.255.255.0
 no shutdown
 exit
ip default-gateway 192.168.1.1
interface range f0/1-2
 switchport mode trunk
 channel-group 1 mode active
 no shutdown
 exit
interface f0/23
 switchport mode access
 spanning-tree portfast
 no shutdown
```

```
 exit
ntp server 192.168.1.1
end
```

b. Set the clock on each device to UTC time.

c. Save the running configuration to startup-config.

d. Configure IPv4 and IPv6 addresses on hosts PC1 and PC2 as shown in the addressing table.

e. Verify that R1, D1, A1, and PC2 can successfully ping PC1 at 192.168.1.50.

Part 2: Configure and Verify Flexible Netflow

As previously stated, Flexible Netflow provides the ability to customize traffic analysis parameters. The workflow for Flexible Netflow consists of four steps:

Step 1. Create Flow Records. Flow records define the information to be collected. There are predefined flow records that match the flow caching done by Classic Netflow, or you can configure your own custom flow record to suit your needs.

Step 2. Create Flow Exporter. This defines where compiled statistic information is sent.

Step 3. Create Flow Monitor and associate Flow Records and Flow Exporters with it.

Step 4. Configure the appropriate interface for input or output caching associated with the appropriate Flow Monitor.

In this part of the lab, you will configure Flexible Netflow to send statistical information about R1 interface g0/0/1 to PC1.

Step 1: Create flow records.

a. For our first flow record, we will use the predefined ipv4 original-input flow record. Because it is predefined, there is no configuration necessary.

b. For our second flow record, we will create a custom flow record. Because the first flow record is focused on input traffic, the second will focus on output traffic. Create a flow record named CCNP8-CUSTOM-OUT.

```
R1(config)# flow record CCNP8-CUSTOM-OUT
```

1) Set up the flow record to match ipv4 destination address and transport destination.

```
R1(config-flow-record)# match ipv4 destination address
R1(config-flow-record)# match transport destination-port
```

2) Set up the flow record to collect bytes and packets.

```
R1(config-flow-record)# collect counter bytes
R1(config-flow-record)# collect counter packets
```

Step 2: Create a flow exporter.

a. The flow exporter configuration defines where the cached information will be sent. Create a flow exporter named CCNP8-COLLECTOR-HOST. Further specify that the exporter should use Netflow version 9, and point to 192.168.1.50 udp port 9996.

```
R1(config)# flow exporter CCNP8-COLLECTOR-HOST
R1(config-flow-exporter)# destination 192.168.1.50
R1(config-flow-exporter)# export-protocol netflow-v9
R1(config-flow-exporter)# transport UDP 9996
R1(config-flow-exporter)# exit
```

Step 3: Create flow monitors.

The flow monitor associates a flow record with the flow exporter. For our exercise, we need to create two flow monitors, one for each flow record.

a. Create the first flow monitor and name it CCNP8-INBOUND-MONITOR using the **flow monitor CCNP8-INBOUND-MONITOR** command. As part of the flow monitor, specify that it will record the netflow ipv4 original-input flow record, export the cache to the exporter every 30 seconds, and identify CCNP8-COLLECTOR-HOST as the exporter.

```
R1(config)# flow monitor CCNP8-INBOUND-MONITOR
R1(config-flow-monitor)# record netflow-original
R1(config-flow-monitor)# exporter CCNP8-COLLECTOR-HOST
R1(config-flow-monitor)# exit
```

b. Create the second flow monitor and name it CCNP8-OUTBOUND-MONITOR using the **flow monitor CCNP8-OUTBOUND-MONITOR** command. As part of the flow monitor, specify that it will record the CCNP8-CUSTOM-OUT flow record, export the cache to the exporter every 30 seconds, and identify CCNP8-COLLECTOR-HOST as the exporter.

```
R1(config)# flow monitor CCNP8-OUTBOUND-MONITOR
R1(config-flow-monitor)# record CCNP8-CUSTOM-OUT
R1(config-flow-monitor)# exporter CCNP8-COLLECTOR-HOST
R1(config-flow-monitor)# exit
```

c. Use the **show flow monitor** command to examine the results.

```
R1# show flow monitor
Flow Monitor CCNP8-INBOUND-MONITOR:
 Description: User defined
 Flow Record: netflow-original
 Flow Exporter: COLLECTOR
 Cache:
 Type: normal
 Status: not allocated
 Size: 4096 entries / 163852 bytes
 Inactive Timeout: 15 seconds
 Active Timeout: 1800 seconds
Update Timeout: 1800 seconds
Flow Monitor CCNP8-OUTBOUND-MONITOR:
 Description: User defined
 Flow Record: CUSTOM-OUT
 Flow Exporter: COLLECTOR
 Cache:
 Type: normal
 Status: not allocated
 Size: 4096 entries / 163852 bytes
 Inactive Timeout: 15 seconds
 Active Timeout: 1800 seconds
 Update Timeout: 1800 seconds
```

Step 1: Configure the interface for flow caching.

The last step is to configure the appropriate interface(s) so that they will cache information. In our lab, we will focus on the input and output from interface g0/0/0 on R1. Use the **ip flow monitor <name> <direction>** command on g0/0/1 to specify the inbound and outbound flow monitors you have created.

```
R1(config)# interface g0/0/1
R1(config-if)# ip flow monitor CCNP8-INBOUND-MONITOR input
```

```
R1(config-if)# ip flow monitor CCNP8-OUTBOUND-MONITOR output
R1(config-if)# exit
```

Step 2: Create some traffic.

To gather statistics, we will need some traffic.

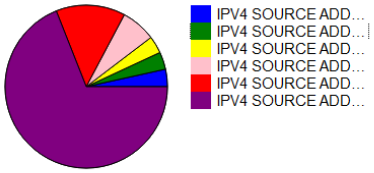
From PC2, start a continuous ping to R1 using IPv4.

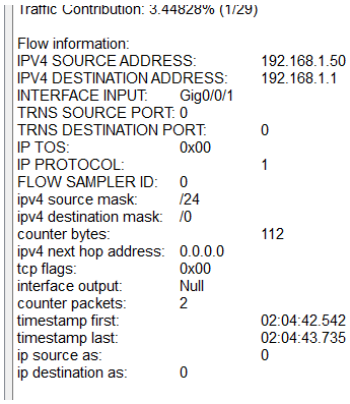
a. From switch A1, telnet to R1. Login and leave the session running.

b. On PC1, start the netflow collector from the link on ye desktop. Enable the application.

Step 3: Wait 60 seconds then examine the results.

a. On PC1, observe the flows entering the application:





b. On R1, issue the command **show flow monitor CCNP8-INBOUND-MONITOR statistics**.

```
R1# show flow monitor CCNP8-INBOUND-MONITOR statistics
  Cache type:                               Normal (Platform cache)
  Cache size:                               200000
  Current entries:                               2
  High Watermark:                               12

  Flows added:                                 103
  Flows aged:                                  101
    - Active timeout      (    30 secs)        38
    - Inactive timeout    (    15 secs)        63
```

c. On R1, issue the command **show flow monitor CCNP8-INBOUND-MONITOR cache**. **Note**: Output will vary depending upon how long within the 30-second window traffic has been caching.

```
R1# show flow monitor CCNP8-INBOUND-MONITOR cache
  Cache type:                               Normal (Platform cache)
  Cache size:                               200000
  Current entries:                               1
  High Watermark:                               12

  Flows added:                                 112
  Flows aged:                                  111
    - Active timeout      (    30 secs)        43
    - Inactive timeout    (    15 secs)        68
```

```
IPV4 SOURCE ADDRESS:        192.168.1.75
IPV4 DESTINATION ADDRESS:   10.0.0.1
TRNS SOURCE PORT:           0
TRNS DESTINATION PORT:      2048
INTERFACE INPUT:            Gi0/0/1
FLOW SAMPLER ID:            0
IP TOS:                     0x00
IP PROTOCOL:                1
ip source as:               0
ip destination as:          0
ipv4 next hop address:      0.0.0.0
ipv4 source mask:           /0
ipv4 destination mask:      /0
tcp flags:                  0x00
interface output:           Null
counter bytes:              12024
counter packets:            8
timestamp first:            20:43:34.189
timestamp last:             20:43:41.263
```

d. Stop all the pings and exit the telnet session.