

# Monitoring Netflow with NfSen - Network Monitoring and Management

## Introduction

### Goals

- Learn how to export flows from your bdr1.campusY router

### Notes

- Commands preceded with “\$” imply that you should execute the command as a general user - not as root.
- Commands preceded with “#” imply that you should be working as root.
- Commands with more specific command lines (e.g. “bdr1.campusY>” or “mysql>”) imply that you are executing commands on remote equipment, or within another program.

## Export flows from bdr1.campusY

You will configure your router to export flows to the database (db) server in your group.

```
$ ssh nmmlab@bdr1.campusY.ws.nsrc.org
bdr1.campusY> enable
```

Now do the following:

```
bdr1.campusY# configure terminal
bdr1.campusY(config)#

flow exporter EXPORTER-1
  description Export to db.campusY
  destination 100.68.Y.130
  transport udp 9996
  template data timeout 60

flow monitor FLOW-MONITOR-V4
  exporter EXPORTER-1
  record netflow ipv4 original-input
  cache timeout active 300

interface FastEthernet 0/0
```

```
ip flow monitor FLOW-MONITOR-V4 input
ip flow monitor FLOW-MONITOR-V4 output
```

```
snmp-server ifindex persist
```

Since you have not specified a protocol version for the exported flow records, you get the default which is Netflow v9.

Netflow v9 packets cannot be decoded by the receiver until it has received a template packet. The command `template data timeout 60` tells the router to send it every 60 seconds, to make the lab exercises work more quickly. (In production a value of 300 is fine).

The `cache timeout active 300` command breaks up long-lived flows into 5-minute fragments. If you leave it at the default of 30 minutes your traffic graphs will have spikes.

Aside: if you want to monitor IPv6 flows you would have to create a new flow monitor for IPv6 and attach it to the interface and the existing exporters.

```
flow monitor FLOW-MONITOR-V6
  exporter EXPORTER-1
  record netflow ipv6 original-input
  cache timeout active 300
interface FastEthernet 0/0
  ipv6 flow monitor FLOW-MONITOR-V6 input
  ipv6 flow monitor FLOW-MONITOR-V6 output
```

The command `snmp-server ifindex persist` enables ifIndex persistence globally. This ensures that the ifIndex values are retained during router reboots - also if you add or remove interface modules to your network devices.

Now we'll verify what we've done.

First exit from the configuration session:

```
bdr1.campusY(config)# end
bdr1.campusY# show flow exporter EXPORTER-1
bdr1.campusY# show flow exporter EXPORTER-2
etc...
bdr1.campusY# show flow monitor FLOW-MONITOR-V4
```

It's possible to see the individual flows that are active in the router:

```
bdr1.campusY# show flow monitor FLOW-MONITOR-V4 cache
```

But on a busy router there will be thousands of individual flows, so that's not useful. Press 'q' to escape from the screen output if necessary.

Instead, group the flows so you can see your "top talkers" (traffic destinations and sources). This is one very long command line:

```
bdr1.campusY# show flow monitor FLOW-MONITOR-V4 cache aggregate ipv4 source address ipv4 des
```

If it all looks good then write your running-config to non-volatile RAM (i.e. the startup-config):

```
bdr1.campusY# wr mem
```

You can exit from the router now:

```
bdr1.campusY# exit
```

To check flow packets are arriving at your VM you can use tcpdump:

Have one person in your group connect to your db-campusY server and then do:

```
$ sudo apt-get install tcpdump
$ sudo tcpdump -i eth0 -nn udp port 9996
```

Wait a few seconds and you should see packets arriving. These are the UDP packets containing individual flow records. After seeing some packets you can press ctrl-c to exit from tcpdump.

You should see something like:

```
sysadm@db:~$ sudo tcpdump -i eth0 -nn -Tcnfp port 9996
[sudo] password for sysadm:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
15:35:10.300755 IP 100.68.0.22.63721 > 100.68.6.130.9996: NetFlow v9
15:35:21.327925 IP 100.68.0.22.63721 > 100.68.6.130.9996: NetFlow v9
15:35:24.303433 IP 100.68.0.22.63721 > 100.68.6.130.9996: NetFlow v9
15:35:28.301635 IP 100.68.0.22.63721 > 100.68.6.130.9996: NetFlow v9
15:35:29.318562 IP 100.68.0.22.63721 > 100.68.6.130.9996: NetFlow v9
15:35:36.312071 IP 100.68.0.22.63721 > 100.68.6.130.9996: NetFlow v9
15:35:38.320587 IP 100.68.0.22.63721 > 100.68.6.130.9996: NetFlow v9
15:35:41.312730 IP 100.68.0.22.63721 > 100.68.6.130.9996: NetFlow v9
15:35:43.305899 IP 100.68.0.22.63721 > 100.68.6.130.9996: NetFlow v9
15:35:48.314821 IP 100.68.0.22.63721 > 100.68.6.130.9996: NetFlow v9
15:35:49.319415 IP 100.68.0.22.63721 > 100.68.6.130.9996: NetFlow v9
15:35:54.290466 IP 100.68.0.22.63721 > 100.68.6.130.9996: NetFlow v9
15:36:02.320703 IP 100.68.0.22.63721 > 100.68.6.130.9996: NetFlow v9
15:36:03.307909 IP 100.68.0.22.63721 > 100.68.6.130.9996: NetFlow v9
15:36:04.320107 IP 100.68.0.22.63721 > 100.68.6.130.9996: NetFlow v9
15:36:18.309020 IP 100.68.0.22.63721 > 100.68.6.130.9996: NetFlow v9
15:36:21.290958 IP 100.68.0.22.63721 > 100.68.6.130.9996: NetFlow v9
^C
17 packets captured
17 packets received by filter
0 packets dropped by kernel
```

Once you see that records are arriving you can log off both machines by doing:

```
$ exit
```

OPTIONAL: you can use tshark (the text version of wireshark), which is able to fully decode Netflow v9 packets:

```
$ sudo apt-get install tshark
```

```
$ sudo tshark -i eth0 -nnV -s0 -d udp.port==9996,cflow udp port 9996
```

You are now done with this lab.