Generating Netflow Traces for Network Configurations

Ominike Akpovi A.
Information & Communications Technology Department
Petroleum Training Institute (PTI)
Nigeria

Abstract— Network performance management is essential to achieve a robust and efficient network. We can anticipate how a network performs if we have enough statistical data about the devices on the network and if we understand how traffic flows between these devices. There are various tools and protocols used for network performance management such as the Simple Network Management Protocol (SNMP) and Common Management Information Protocol (CMIP). This paper is aimed at using the more advanced netflow protocol, which has the ability to characterize IP traffic, to generate netflow traces, which will be archived for the purpose of network performance analysis.

Index Terms— Network Performance, Netflow, Network Monitoring.

1 Introduction

The primary components of any networked computing environment are the computing and communication resources that make up the enterprise network. Today's networks are multi-vendor in their design because they contain network elements produced by many vendors. To successfully operate and manage a network, it is important that network management is taken into consideration in the early planning and design stages. In very small networks with low usage the efforts by network managers to successfully administer the network and its resources may be minimal, but as networks grow in complexity, network management tools and techniques become indispensable to effectively and efficiently run the network [3]. Organizations, institutions and businesses today are increasingly dependent on networked services and resources and the efficiency of these network services ensure smooth running of the organizations. With efficient network and traffic monitoring, we can discover potential bottlenecks, increased traffic utilization and feedback on network performance, which will help us to provide quality network services in a cost effective way. Until recently, most networks relied almost exclusively on Simple Network Management Protocol (SNMP) to monitor performance and bandwidth utilization.

SNMP, however, is not very effective in characterizing traffic applications and patterns, essential for understanding the performance of a network. To gain a more granular understanding of network performance, the netflow protocol was created. Netflow is an embedded instrumentation within Cisco internetwork operating system (IOS) Software used to characterize network operations. Network administrators need visibility into the network to understand how the network is behaving. Netflow provides network administrators with comprehensive information about how network traffic flows [4].

With such knowledge of network behaviour, business processes will improve and networks will be operated more efficiently, which will in turn, lower operational costs and improve business revenues by better utilization of the network infrastructure. It is imperative to mention that although netflow is a Cisco proprietary protocol, it works well with other devices that understand it. A large number of router/switch vendors support the netflow format. They support distinct "flow data" formats that fall under the de-facto term netflow [5]. In this paper, the focus is on using open source / freeware to generate the netflow (i.e. to act as the Cisco device).

2 Network Performance Management

To successfully manage a network, it is important that performance management is given priority. This involves collecting data about how the network elements are behaving, understanding the traffic flow patterns that are occurring in the network at different periods of activity and subsequently being able to predict how the network will perform [1]. To be able to offer efficient and reliable services in computer networks, we need to be able to effectively monitor network parameters, relevant traffic and infer from these measurements relevant information [2]. With efficient network and traffic monitoring, we will be able to discover potential bottlenecks, increased traffic utilization and feedback on network performance, which will help us to provide quality network services in a cost effective way.

At the most basic level, the idea is to get a snapshot of the current performance. However, to obtain a more detailed analysis, some network parameters might need to be monitored over a period of time. For example, you might want to plot a graph showing the relationship between some performance values, over a particular time interval. In this time period, you can observe how the value of the parameter changes with time and thus accurately decipher a sudden drop or spike in value out of the ordinary. Different patterns will lead you to anticipate different scenarios. For instance, an unusual increase in

• *Akpovi Ominike is currently pursuing his PhD in Computer Science in Babcock University, Nigeria. E-mail: aominike@yahoo.com*

International Journal of Scientific & Engineering Research Volume 7, Issue 5, May-2016
ISSN 2229-5518

utilization on an Ethernet interface might mean a higher number of dropped packets which might ultimately lead to application sessions timing out. When you monitor network performance, you anticipate, see and resolve problems before they happen. With careful observation, when you observe values over a period of time, you can identify patterns, bottlenecks, network pain-points and unusual network behaviour. Armed with such invaluable information about your network, network modification and configuration changes can be carried out to optimize the network [7]. We should be interested in collecting performance data from our network, which can be provided as a stream of events using protocols e.g. netflow

3 NETFLOW

It is important to understand how a network performs. This can be achieved by characterizing traffic patterns, IP traffic and by monitoring bandwidth utilization. In the past, the main tool used by network managers to monitor network performance and bandwidth utilization was the Simple Network Management Protocol (SNMP). SNMP was not very effective at doing this and thus there was the need for a more robust system. Netflow has the ability to characterize IP traffic and this is important when monitoring network availability and performance and during troubleshooting. When IP traffic flows are monitored, it allows for efficient resource utilization and more accurate network capacity planning. Quality of Service (QoS) and Network security is improved, as network managers are able to identify the exact pain-points, detect attacks and other vulnerabilities [4]. Netflow works like this: when the router receives a packet, its netflow module scans for key details like the source and destination IP address, the source and destination port number, the protocol type, the type of service (ToS) bit in the IP header, and the interface number on the router of the IP packet to determine if the packet is part of an already existing flow. If the flow already exists, it updates that flow record; else, a new flow record is created [6].

An IP flow is a unidirectional stream of packets that pass through a network element and share a common set of attributes [8], [9]. Fig.1 shows the attributes of a packet, which is used to create the flow. Every packet is examined for a set of attributes. These attributes constitute the fingerprint of the packet, which explains if a packet is unique, or not. Often times, an IP Flow consists of between five and seven packet attributes namely- IP source and destination address, Source and Destination port, Protocol type, Class of Service and Router/Switch interface. Netflow records provide great insights into understanding how networks behave. The interface defines how the network device utilizes traffic, the class of service (CoS) identifies the priority of the traffic, the source and destination address defines the origin and target of the traffic respectively while the source and destination port numbers tell us the applications utilizing the traffic

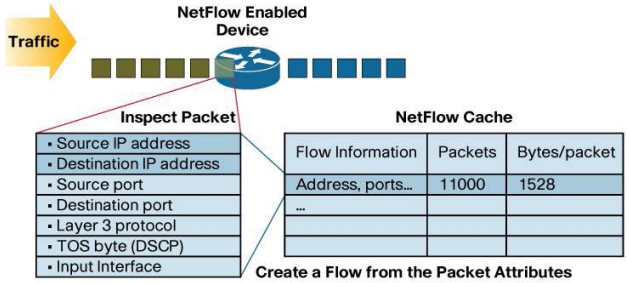

Figure 1: Creating a flow in the netflow cache [4].

Netflow facilitates solutions to many common problems encountered by IT professionals. The major benefits of netflow are that it enables us to: analyze applications and their impact on the network, measure WAN traffic i.e. understand who is utilizing the network and the network top talkers, troubleshoot and understand network pain points, detect unauthorized traffic and prevent expensive (sometimes unrequired) upgrades by identifying the applications causing congestion and it enables us validate QoS parameters [4]. Netflow has undergone several revisions through its history and it exports data flow information in different formats ranging from netflow version 1 to version 10.

4 OPEN SOURCE NETFLOW ARCHITECTURE

The major components of an open source netflow system are the sensor, the collector, and some sort of reporting system. The collector and sensors are the irreplaceable components of any flow system because you can analyze data in innumerable ways, but before you do, you must gather and store the data, and you do that with sensors and the collector [12]. Fig. 2 illustrates the netflow architecture with a netflow sensor and netflow collector.

The sensor (otherwise called a probe or generator) is a daemon that listens to the network and captures your session data. The sensor sends the session information to the collector. A good place to set up the sensor is in the network Ethernet core. This is because the internal LAN has a lot of traffic and analyzing flow data from the internal network will quickly expose problems, wrong configurations and performance issues [12]. For this research work, flowprobe also known as fprobe was used as the network sensor. Flowprobe exports captured traffic to a remote netflow collector. It captures network traffic and emits it as netflow to a specified collector [11].

The collector listens for reports from the sensor on a selected UDP port of your choice and saves them in a file for later evaluation. The collector needs to connect to a hub, mirrored switch port, or any device where it can see all the network traffic [10]. Flow collection uses very few system resources other than disk space and the amount of disk space needed depends on the type of traffic on your network and how long you want to retain your records. However, additional memory and CPU resources will accelerate flow reporting [12].

International Journal of Scientific & Engineering Research Volume 7, Issue 5, May-2016
ISSN 2229-5518

In this research work I used the netflow capture daemon, nfcapd and nfdump for the collector software.
Nfcapd and Nfdump work hand in hand. Nfdump reads the netflow data from files stored by nfcapd processes the data and produces results according the options selected.
In this research work, a network testbed was set up using PC's with Ubuntu Linux operating system to run the netflow sensor and collector software. This choice was due to the fact that Ubuntu Linux belongs to the open source community and it has majority of the tools required for netflow implementation. Finally, the reporting system reads the files generated by the collector and produces reports in text files, which are easily readable.

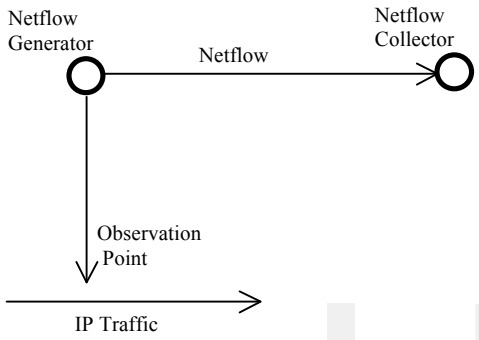

Figure 2: Netflow Architecture With Netflow Sensor and Collector

5 Implementation

To generate the netflow traces, different network test bed configurations were set up. In Fig. 3, a sample network configuration scenario used to generate netflow traces is shown. I installed and configured fprobe and then configured the netflow sensor to generate netflow records by issuing the following command at the terminal window: "fprobe –i eth2 192.168.0.12:55000". This means that the sensor fprobe should store all data transfers on the Ethernet 2 interface and send the collected data to the host with IP address 192.168.0.12 via UDP port 55000. I configured the address and port of the netflow collector as 192.168.0.12 and 55000 respectively by issuing the following command at the terminal window: "nfcapd -b 192.168.0.12 –p 55000 -l/home/Akpovi/netflow" This means that the netflow collector should listen to the UDP port 55000, capture the netflow data it sees and save the data as files in the /home/Akpovi/netflow folder. The captured netflow files (nfcapd) and netflow records were archived in a netflow folder created on the collector PC (192.168.0.12).

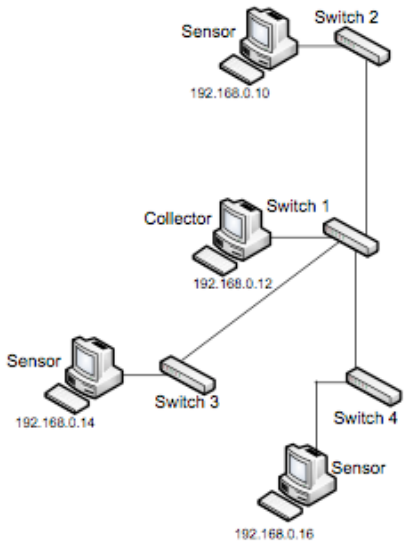

Figure 3: Sample Network Configuration Used to Generate Netflow traces.

6 Results and Analysis

A basic analysis of the results generated from the star configuration is discussed below. Fig. 4 shows the netflow records of host 192.168.0.10 from April 10, 2016 at 12:45. Each line listing in the netflow records as shown in the figure represents one flow. The first column, the Date flow start, shows the date and the time the flow was started and the second column (Duration) shows the duration of the flow. The date and Duration of the flow are given in millisecond resolution. The Proto column displays the protocol type for this flow. The Src IP Addr: Port column shows the source IP address of the flow and the source port, while the Dst IP Addr: Port column shows the destination address and port. Finally, the packets column shows the number of packets in the flow and the Bytes column shows the number of bytes in the flow. At the end of the flows a summary listing is shown which contains information about the total flows, total bytes, total packets, average packet per second (avg pps), average bits per second (avg bps) and the average bytes per packet (avg bpp). From Fig. 4, we can noticeably observe that between 12:42 and 12:43, there was a significant increase in the traffic sent between hosts 192.168.0.10 and 192.168.0.12.This was as a result of a media streaming activity going on between the two hosts, using http port 8080.

The great news about netflow is that we have actual data about our network, but the not so good news is that we have far too much data about the network. As network size and complexity increases, the number of flows generated will also increase. To manage this, we must filter our data to display only interesting flows. This will help us evaluate and diagnose network issues. Netflow can be filtered in various ways to investigate network behaviour by using primitives such as Proto (protocol), Src ip (source ip), Dst ip (destination ip), Packets and Bandwidth.

International Journal of Scientific & Engineering Research Volume 7, Issue 5, May-2016
ISSN 2229-5518

```
akpovi2@akpovi2-HP-Compaq-dx2300-Microtower:~$ nfdump -r/home/netflow/nfcapd.201208101246 'host 192.168.0.10'

Date flow start          Duration Proto      Src IP Addr:Port          Dst IP Addr:Port   Packets    Bytes Flows
2016-04-10 12:45:52.380    5.000 UDP       192.168.0.10:34346 ->     192.168.0.12:55000       2      248     1
2016-04-10 12:46:26.424    0.000 UDP       192.168.0.16:49287 ->     192.168.0.10:30712       1      148     1
2016-04-10 12:46:21.000    5.000 UDP       192.168.0.10:34346 ->     192.168.0.12:55000       2      248     1
2016-04-10 12:46:26.424    0.000 ICMP      192.168.0.10:0     ->     192.168.0.16:3.0         1      176     1
2016-04-10 12:42:37.585  303.349 TCP       192.168.0.10:8080  ->     192.168.0.12:40654    12656   16.6 M     1
2016-04-10 12:42:37.585  303.387 TCP       192.168.0.12:40654 ->     192.168.0.10:8080      5229   272209     1
2016-04-10 12:43:06.215  304.749 TCP       192.168.0.10:8080  ->     192.168.0.12:40654    13445   16.8 M     1
2016-04-10 12:43:06.215  304.751 TCP       192.168.0.12:40654 ->     192.168.0.10:8080      5254   273509     1
2016-04-10 12:47:23.028    0.000 UDP       192.168.0.10:42144 ->    192.168.0.255:50000       1      167     1
2016-04-10 12:47:51.641    0.499 TCP       192.168.0.10:43042 ->     192.168.0.14:50000       2      290     1
2016-04-10 12:47:51.640    0.000 UDP       192.168.0.10:42144 ->    192.168.0.255:50000       1      167     1
2016-04-10 12:47:51.643    0.461 TCP       192.168.0.14:50000 ->     192.168.0.10:43042       2      338     1
2016-04-10 12:47:23.027    5.272 TCP       192.168.0.10:33180 ->     192.168.0.12:50000       4      628     1
2016-04-10 12:47:23.028    5.308 TCP       192.168.0.12:50000 ->     192.168.0.10:33180       4      628     1
2016-04-10 12:47:51.641    5.272 TCP       192.168.0.10:33180 ->     192.168.0.12:50000       4      628     1
2016-04-10 12:47:51.643    5.308 TCP       192.168.0.12:50000 ->     192.168.0.10:33180       4      628     1
2016-04-10 12:49:00.437    0.000 TCP       192.168.0.10:43042 ->     192.168.0.14:50000       2      338     1
2016-04-10 12:49:00.437    0.000 TCP       192.168.0.14:50000 ->     192.168.0.10:43042       2      290     1
2016-04-10 12:48:11.266   55.327 UDP       192.168.0.16:49287 ->     192.168.0.10:30712       2      248     1
2016-04-10 12:48:11.266   55.327 ICMP      192.168.0.10:0     ->     192.168.0.16:3.0         2      304     1
2016-04-10 12:50:12.560    0.000 UDP       192.168.0.16:49287 ->     192.168.0.10:30712       1      148     1
2016-04-10 12:50:12.560    0.000 ICMP      192.168.0.10:0     ->     192.168.0.16:3.0         1      176     1
2016-04-10 12:47:07.384  160.009 UDP       192.168.0.10:34346 ->     192.168.0.12:55000       6     1128     1
2016-04-10 12:47:36.000  160.000 UDP       192.168.0.10:34346 ->     192.168.0.12:55000       6     1128     1
2016-04-10 12:47:13.893  194.745 TCP       192.168.0.16:41120 ->     192.168.0.10:80       48975   2.6 M     1
2016-04-10 12:47:13.893  194.745 TCP       192.168.0.10:80    ->     192.168.0.16:41120    98430  147.6 M     1
Summary: total flows: 26, total bytes: 184.2 M, total packets: 184039, avg bps: 3.1 M, avg pps: 390, avg bpp: 1000
Time window: 2016-04-10 12:42:37 - 2016-04-10 12:50:28
Total flows processed: 36, Blocks skipped: 0, Bytes read: 1900
Sys: 0.000s flows/second: 0.0        Wall: 0.000s flows/second: 90909.1
akpovi2@akpovi2-HP-Compaq-dx2300-Microtower:~$
```

Figure 4: Screenshot of netflow records

7 Conclusion and Future Work

Networks today are very complex and dynamic and as networks have evolved, the tendency has been to promote increased convergence of voice and video services onto traditional data networks. As a result of these, it has become imperative to have clear and distinct visibility into network performance and this is exactly what flow monitoring data does. Experiments were conducted by installing and running different applications of varying memory and processor utilization and the netflow records were collected for different network configurations. The netflow capture daemon rotates the captured netflow files at a default time interval of 300s (5min). However, this default time can be increased to reduce the frequency at which the files are captured, which will ultimately save disk space.

From the results obtained it can be seen that the netflow traces gave application-level details such as hosts communicating, protocols being used, packet size and byte size, which are an inherent part of IP traffic and which will offer in-depth and fine-grained bandwidth analysis. The choice of applications used to generate the netflow traces were limited to those that could run over a local area network (LAN). In future, to obtain more robust netflow records, real world data could be obtained using a SPAN port. Also, in future research work, a graphical web based utility could be designed to give a graphical overview of the netflow data in addition to using the command line interface.

References

[1] A. King, and R. Hunt, "Protocols and architecture for managing TCP/IP network infrastructures," *Computer Communications*, Volume 23, Issue 2, Pages 1558–1572, March 2000.

[2] D.Raz, "Efficient Network and Traffic Monitoring," *IEEE Network Operations and Management Symposium*, Page(s): 587 - 587, April 2006.

[3] Gilbert Held, 1998. *Ethernet Networks.* USA, John Wiley & Sons, Inc. Third Edition.

[4] Cisco Systems, Inc. (2012) "Introduction to Cisco IOS® Netflow", http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6555/ps6601/prod_white_paper0900aecd80406232.pdf [viewed 15th April 2016].

[5] Xangati Inc., (2008) "Frequently Asked Questions on Netflow", http://www.xangati.com/documents/FAQforNetFlow..pdf [viewed 15th April 2016].

[6] Liu Bin, Lin Chuang, Qiao Jian, He Jianping and Peter Ungsunan, "A NetFlow based flow analysis and monitoring system in enterprise", *Computer Communications*, Volume 52, Issue 2, Pages 1074-1092, January 2008.

[7] Alexander Clemm, 2007. *Network Management Fundamentals.* USA, Cisco Press, Cisco systems, Inc.

[8] I. Drago, R. Rafael, R. Barbosa, R.Sadre, A.Pras and J. Schonwalder, "Report of the Second Workshop on the Usage of NetFlow/IPFIX in Network Management", *J Netw Syst Manage,* Volume 19, Issue 2, Pages 298–304, March 2010.

[9] A. Puliafito and O. Tomarchio, "Using mobile agents to implement flexible network management strategies," *Computer Communications,* Volume 23, Pages 708-719, 2000.

[10] O'Reilly Media Inc., (2005) "Monitoring Network Traffic with Netflow", http://onlamp.com/pub/a/bsd/2005/08/18/Big_Scary_Daemons.html [viewed 8th March 2016]

[11] Canonical Ltd, (2012) "Binary package "fprobe" in Ubuntu lucid" https://launchpad.net/ubuntu/lucid/+package/fprobe [viewed 8th April 2016]

[12] Michael W. Lucas, 2010. *Network Flow Analysis.* No Starch Press, Inc.