LUWEMBA MUSA

Practical analysis of flows with IPFIX

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Thesis
10 May 2014

Helsinki Metropolia University of Applied Sciences

Author(s) Title Number of Pages Date	Luwemba Musa Practical analysis of flows with IPFIX 35 pages + 6 appendices 10 May 2014
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Networking
Instructor(s)	Matti Puska, M. Sc, Principal Lecturer

The goal of the project was to implement and analyze the flow table update using the IP-FIX protocol. A network flow monitoring system is set up in away where the devices to be monitored send packets to a collector that analyses these packets and statistical graphs and charts are generated and displayed on a web page.

A shortlist of software was reviewed and three of the best working conditions were used to analyze the protocol. Each software has its advantages and drawbacks. The similarities and differences are reported in the report.

A network was designed with three Cisco routers and a switch and three computers connected to the switch. The computers acted as the collectors of the packets and hosted the software while the routers were the Agents that sent the packets to the computers. Traffic was generated in the network and the packets were collected and analyzed.

The collected packets, after being analyzed could clearly show the network design, how it is being utilized with graphs, pie charts and log messages that are displayed on a web page of each software type. The results obtained from the project are convincing and give a clear picture of the monitored network with all the three different types of software.

Keywords	IPFIX, NetFlow, SNMP, packets, data, collector, agent.

Contents

1 Introduction

This chapter serves as an introduction to the topic of the thesis and gives an overview of the ideas and motivations behind the topic.

NetFlow fulfils the needs to create an environment where the administrators have the tools to understand who, what, when, where and how network traffic is flowing. Network operators and designers take into account the type of traffic that will be flowing through the network and the distribution of the network capacity with volume of the traffic so the need to measure the traffic is paramount in any given network design in order to have a balance between requirement and availability. Network operators also need a detailed view of the network traffic for security reasons. [1]

The network analysis is done by finding out the most used applications or the heaviest users on the network. Traffic measurement can be done by analyzing the loggings from packets passing through each network device (router, switch) however with the increasingly high volumes of data brought about with the growing network utilization, this strategy is no longer feasible. Instead packets with the same properties are grouped together in to flows which keep statistic record of the traffic they are generated from. In this way similar types of traffic can be stored in a general format without losing of data.

Networks built on Internet Protocol (IP) are not designed to reveal detailed statistics of the traffic between two endpoints, making it hard to measure IP network traffic. Very few measurement capabilities are enabled in different protocols operating on different network layers, thus the challenges surrounding the study of measurements in IP network traffic. The newest technology developed for network traffic monitoring is Internet Protocol Flow Information Export (IPFIX). [2]

The IPFIX is an Internet Engineering Task Force (IETF) protocol, as well as name of the working group defining the protocol. It was created based on a need for a common, universal standard of export for Internet Protocol flow information from network devices such as routers and switches, to facilitate services such as measuring, account-

ing and billing network traffic. The IPFIX technology defines the IP flow information to be formatted and transferred from an exporter to a collector agent. [3, 2]

As recently as 2000, Cisco was only using the Simple Network Management Protocol (SNMP) to manage the network. This could work but could not characterize the traffic. Cisco implemented NetFlow that could characterize the traffic, a protocol that had been built in 1996. In 2003 NetFlow version 9 was standardized with the Internet Engineering Task Force (IETF) and the standardized protocol was named IPFIX. [2]

There are different types of software that have been designed to implement the analysis of traffic from the exporter to the collector agent using IPFIX. The collector agent hosts the software with pre-configured parameters that much with the parameters that are configured on the exporters so that the exporters can send the flows to the right collectors. These parameters can either be port numbers or an SNMP community name or source and destination addresses. The exporter device is given the IP address of the collector and the type of flows to be sent.

The goal of the project is to practically build a network with Cisco routers and a switch, install each type of software on a different computer to do the collecting of traffic and generate the traffic with either Ping or Hypertext Transfer Protocol (HTTP). The traffic generated and collected will then be analyzed using the IPFIX protocol.

2 Theoretical background

This chapter discusses what is already known about the IPFIX, and how it has evolved from network management and other forms of traffic flows that are being used.

2.1 Network Management

Network management is the task to control or predict network faults in order to act accordingly to avoid them and to maintain the performance requirement standards. These include activities, methods, procedures and tools that are used to maintain, manage, account and provision a network. Network faults can be caused by various reasons with a big source of faults caused by human errors (users, network engineers, system administrators). Other sources of faults include technical systems cabling, software bugs and configurations. Reliability can be made better by improving network design and fault tolerance solutions. [4]

2.1.1 Network Availability

Network reliability is the measure of its availability when needed. In statistics, availability can be calculated with this formula:

Availability = $\left(\frac{MTBF}{MTBF+MTTR}\right) * 100\%$ [4]

where MTBF is mean time between failures
MTTR is mean time to repair.

Network performance is also very important to users which can be seen with the network response of a desired service. Network Management System (NMS) is defined as a set of technical solutions, which helps network engineers to ensure network availability and usability after a system set-up [5, 2]. This helps in an early identification of the fault, correcting it before it is noticed by users. After a fault has occurred, the time it takes to repair the damages is divided into the following time components below:

- Detection time
- Response time
- Repair time
- Recovery time [5, 8].

2.1.2 Accounting Management

Accounting Management is the process of calculating the resource usage of the network by an individual or group. This is important in the charge back by the operator while billing the users. The measure of the usage of the network resources can be done using NetFlow or IP Accounting features. [6]

Accounting Management can be used fully in capacity planning and network design and it can also controls the quality of services given out by the operator using the access control lists in IP Accounting.

2.1.3 Configuration Management

Configuration Management is the control of the configurations of the system, the devices and the hardware of the network components of that particular system in order for the system to perform as intended. Configuration Management also helps in the documentation and storage of the configuration files and updates. Configuration management helps with network consistence, documents the changes that are made on the network and also defines the best configurations to the network with both security and quality in mind. [7]

2.1.4 Network Management Components

The architecture used by all network management systems is the client/ server architecture. It is shown in figure 1 below.

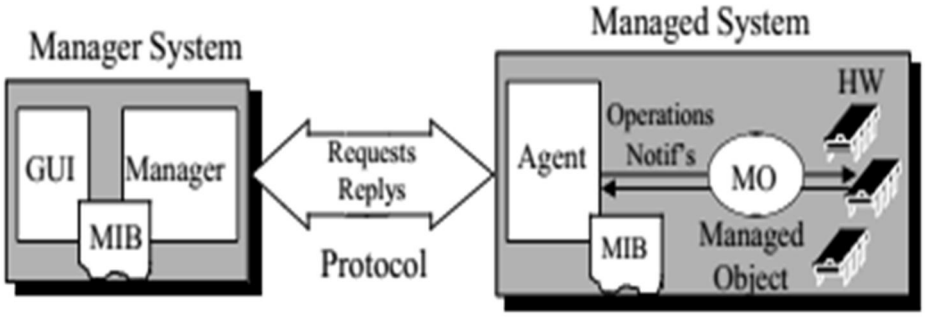


Figure 1: NMS architecture. Copied from Network Management in IP Networks [5, 19].

As shown in figure 1 above, the system includes the following components;
The Manager is a component that the network engineer uses to monitor the whole network system, its performance and the events taking place at a specified time interval. This machine sends requests to pre-configured devices and these devices respond with information that the manager displays to the user interface. The pre-configured devices can also send information that is not requested according to their configuration to the manager. Since the manager offers user interface, it is called the NMS Client. [5, 19]

An Agent is a software component on a managed network device. It acts on behalf of the managed devices by receiving the Client requests, understands the request and gives a desired reply to the manager. [5,19] Clients and agents communicate with either an in-band management or through a separate network using a special application Protocol for example Simple Network Management Protocol (SNMP) is the TCP/IP NMS protocol. [5, 20] Data model and structure defines the management properties (access rights, agent operations, behaviors of the agent, notification) of the device being managed. Both the manager and the agent must have the same data model and identification scheme. [5, 20]

2.1.5 NMS Protocols

The most common protocols today are the Transmission Control Protocol and Internet Protocol TCP/IP. The TCP/IP protocol for network management is the Simple Network Management Protocol (SNMP) developed in the late 1980s. [5, 19] As the name says, it is a very simple protocol as it uses the User Datagram Protocol (UDP). The SNMP trap messages are sent to UDP port 162 and all other messages are sent to UDP port 16. [5, 19]

The Management Information Base (MIB) is used to collect information of an object on a managed device to the management agent using a management protocol. Every object of an MIB has a unique identifier that is used to distinguish it from other objects. The structure of an MIB is tree-like where similar objects are grouped under one node of the tree. [8]

Remote Network Monitoring (RMON) is a portion of the MIB that facilitates proactive network management. These are probes that are installed on devices and collect information and store it. They can be set to send the information after a certain condition has been met. This will reduce the bandwidth usage of transferring information that is not so critical at a particular moment. [9]

2.1.6 Simple Network Management Protocol (SNMP)

The SNMP is an application protocol developed to manage nodes (servers, workstations, routers, switches and hubs) on an IP/TCP network. Network management systems learn about the events of the network through the receiving notifications from network devices implementing SNMP. [10, 37] Currently there are three versions of the SNMP defined: SNMP V1, SNMP V2and SNMPV3. There are many features common between version 1 and version 2, and version 2 enhances additional protocol operations. SNMP V3 adds security and remote configurations capabilities. [10, 37] The protocol structure of the SNMP consists of three fields listed below. Since SNMP is embroiled in UDP, it is an application protocol.

- Version
- Community
- Protocol Data Unit (PDU)

The Version represents the SNMP version number. In order to have a link between the manager and agent, both the manager and agent must use the same version of SNMP. Massages with mismatching version numbers are discarded without processing. The Community represents a string (name) for authenticating the manager before allowing access to the agent and the PDU types are different for different versions as briefly shown below:
SNMP V1: GetRequest, GetNextRequest, GetResponse, SetRequest and trap [10, 38]
SNMP V2 and V3: Get, Get Next, Inform, Response, Set, Trap, Get bulk [10, 40]

2.2 NetFlow

NetFlow is a Cisco developed network protocol for collecting traffic information and also monitoring traffic. When traffic is analyzed, a clear picture of the network can be formulated by knowing which flow is coming in which direction and where it is going. In response to new network requirements, network engineers find it important to understand the behavior of a network. [1] NetFlow has the ability to characterize traffic which was not feasible with the SNMP. Hence it is a better tool in network monitoring availability and performance.

2.2.1 IP Flow

An IP flow is a packet that is forwarded within a networking device with a set of IP packet attributes. An IP attribute is a packet identity that distinguishes a packet from all other packets hence making it unique. There are seven IP packet attributes listed below used for NetFlow:

-IP source address
-IP destination address
-Source port

-Destination port

-Layer 3 protocol types

-Type of service

-Router or switch interface.

Packets with the same source address, destination address, same source port, destination port, same layer 3 protocol type and the Type Of Service (TOS) are grouped into a flow and the packets are numbered or tagged. This helps in scaling down data that is condensed into the NetFlow Cache [11] stored in the device database. This is shown in figure 2 below.

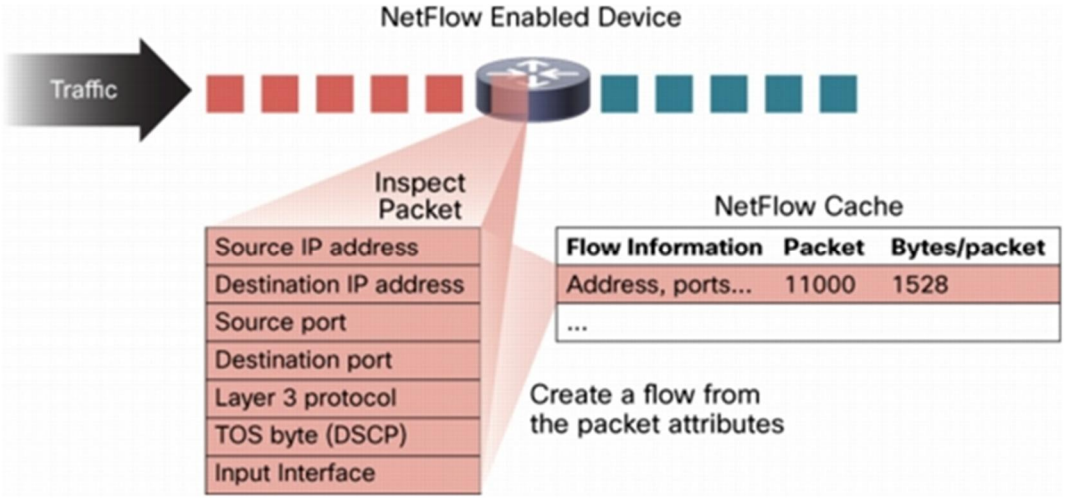


Figure 2: IP traffic flow. Copied from Cisco Network Management System (2007) [6]

The information in figure 2 above is very important in understanding the network characteristics. The source address helps identify where the packets are coming from and the destination address tell where the packets are being received. The ports show which application is being used and the tallied packet shows the amount of traffic collected.

2.2.2 Flow Cache

Devices with the ability to perform layer 4 or layer 5 forwarding functionality have to solve the problem of fast forwarding at the lowest costs. In the Flow cache, only the first packet's header is filtered and the table is matched. The rest of the packets are

forwarded through the cache lookup. As shown in figure 3 below, the Flow Cache takes the fast path.

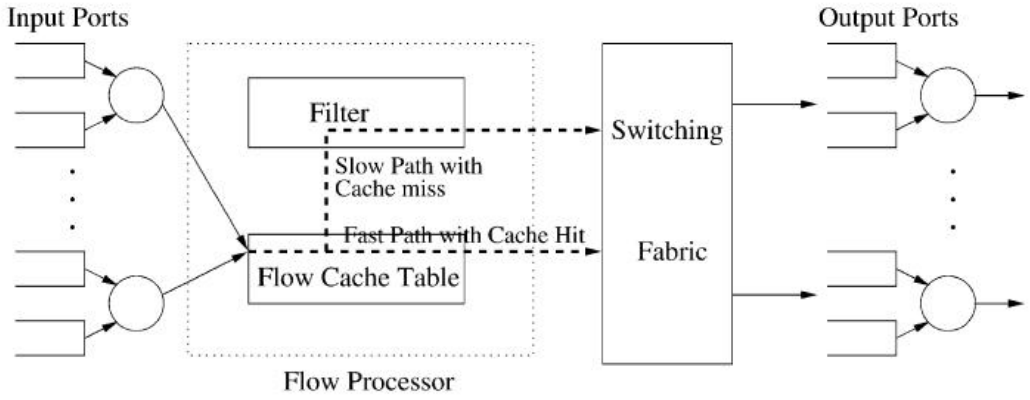


Figure 3: Flow processor. Copied from Ye Tung (2001) [12].

There are other slower flow processes that use the approach of packet by packet header filtering. Every packet must be filtered and then switched through the cache lookup.

2.2.3 NetFlow Access

There are two methods to access the data in the NetFlow cache. One is using the 'show commands' in the command line interface. This is good for troubleshooting the network or wanting to have an immediate view of the network. [1]
The second method is for using an application reporting software that collects the data from the NetFlow cache to a NetFlow collector. The collector has the task to identify and understand the flow being sent. It uses this data to make the traffic reports that help a network engineer.

A NetFlow is ready to be transported to the collector if it is known to be inactive or if the flow has been active for more than a set time to live, for example has been active for longer than the active timer. This is to control the usage of the bandwidth by controlling the amount of information being sent to the collector and to avoid resending the same flow. The default inactive flow timer is 14 seconds while the active time is 30 minutes but all this can be changed in configuration. [1]

2.2.4 NetFlow Versions

Cisco has worked on almost 10 NetFlow versions. Version 2, 3 and 4 were not released. Version 1 (V1) was the first one to be implemented in the initial NetFlow release. Version 5 (V5) is an improvement of version 1 and adds to it Border Gateway Protocol (BGP) autonomous system data and also a flow sequence number. Version 6 (V6) which is similar to version 7 is not supported in the new Cisco IOS release. Version 7 (V7) is only supported in NetFlow with the Cisco Catalyst 5000 series switches equipped with a NetFlow feature card (NFFC). This version is not supported on Cisco Routers. Version 8 (V8) is an improvement of V5 that adds router-based aggregation to it. Version 9 (V9) was developed to support other different technologies such as Multicast, Internet Protocol security (IPsec), Multi-Protocol label Switching. Version 10 is branded as IPFIX. [11]

2.3 Internet Protocol Flow Information Export (IPFIX)

The IPFIX is a protocol developed by the IETF IPFIX working group, whose goal was to standardize the exportation of network data flows from a managed device to the collect agent. The collection of IP traffic mostly consists of IP flows passing through the network element for administrative management purposes. This involves uniqueness in the method of representing the network flow information and the means of communicating the flows from the network elements to the collection point. [3, 3]

2.3.1 Terminology

This section briefly explains some terms commonly used when discussing IPFIX

A flow or IP Traffic flow is a set of IP packets passing an Observation Point in the network during a certain time interval. All packets belonging to a particular flow have a set of common properties. *A packet* is defined to belong to a flow if it completely satisfies all the defined properties of a flow. This ranges from a flow containing all packets seen at a network interface to a flow consisting of just a single packet between two applications such as sampled packets. [3, 4]

Observation Point is a point in a network where IP packets can be observed. Normally this means a central node in a network, typically a router or a core switch. Every Observation Point is associated with an Observation Domain, and that one Observation Point may be a superset of several other Observation Points. [3, 4] *Observation Domain* is the largest set of Observation Points for which flow information can be aggregated by a metering process. The *Flow Record* contains information about a specific flow that was observed at an *Observation Point.* The flow record contains measured properties of the flow and usually characterized properties of the flow. [3, 4]

The Metering Process generates flow records. Inputs to the process are packet headers and characteristics observed at an Observation Point, and the packet treatment at the observation point. The Metering Process consists of a set of functions including packet headers capturing, time stamping, sampling, classifying and maintaining flow records. The maintenance of flow records may include creating new records, updating existing records, computing flow statistics, deriving further flow properties, detecting flow expiration, passing flow records to the exporting process, and deleting flow records. [3, 6]

The Exporting Process is the sending of flow records to one or more collecting processes where records are generated by metering processes while an *Exporter* is a device that hosts one or more exporting processes. The *Collecting Process* receives flow records from one or more exporting processes. The collection process might process or store the received flow records. [3, 7]

The Template is an ordered sequence of <type, length> pair used to completely specify the structure and semantics of a particular set of information that needs to be communicated from an IPFIX device to the collector. Each template is uniquely identifiable by a temp ID. *IPFIX Message* is a message originating at the exporting process that carries the IPFIX records of this exporting process and the destination is a collection process. [3, 7]

2.3.2 IPFIX Architecture

The skeleton of IPFIX is seen in figure 4 below with the exporter directly connected to the network and the collector connected to the exporter with a transportation protocol.

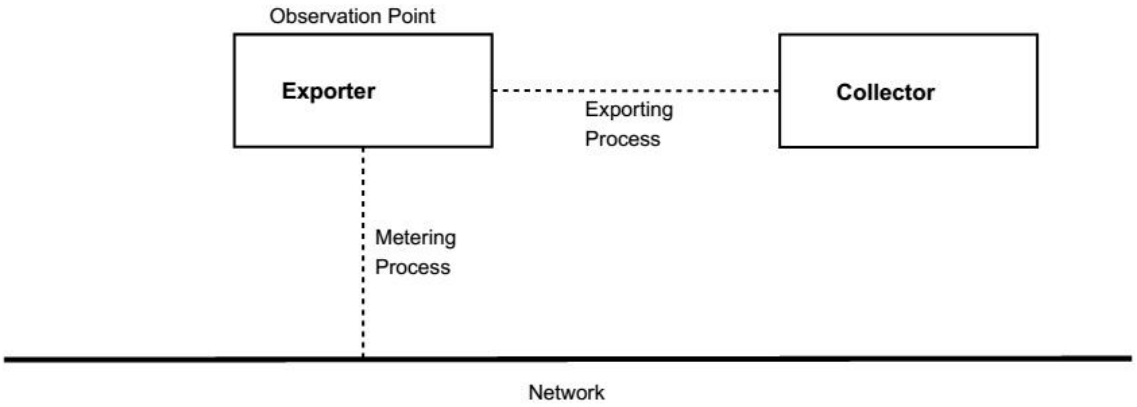


Figure 4: IPFIX Architecture.

A metering process collects data packets at an observation point which can either filter the data or not and then aggregate information about these packets. Using the IPFIX protocol, the information is transported or exported to the collector. Exporters and collectors are in a many-by-many relationship. For example one exporter can send flow packets to many collectors or many exporters can send flow packets to one collector or many exporters can send flow packets to many collectors. [3, 81]

2.3.3 Packet Filtering and sampling

A metering process defines the procedure, so that some packets within the incoming flow are chosen to be analyzed or dropped at an observation point. This can be done by either filtering or sampling or both of them. The order of the sequence does not matter as sampling can come before filtering and also more than one filter can be used in any sequence. The figure 5 below shows the selection criteria of packets using sampling and filtering.

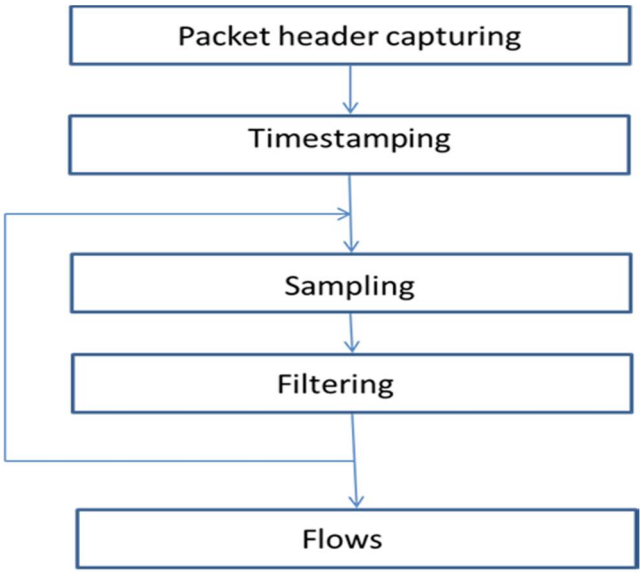


Figure 5: Selection Criteria for packet. Data gathered from Puska 2014 [5, 13]

A sampling function determines which packets with in a flow of incoming packets are selected for measurements. For example a packet that satisfies the sampling limitation are selected to be transported to the collector. This can be every 10th packet received at an observation point. A sample rate of one for every one packet is equal to no sample. A filter function selects only those incoming packets that satisfy a function on fields defined by the packet header fields, fields obtained while doing the packet processing or properties of the packet itself forexample mask match, or port number. [5, 14]

2.3.4 IPFIX Message

IPFIX is a push protocol. The devices that are managed will periodically push IPFIX messages to the configured collector without any intervention by the receiver. Therefore most of the contents in an IPFIX message is to a great extent up to the sender. IPFIX introduces the content of the messages to the collector with the help of special templates. The receiver is free to use or adapt to these sender-defined messages making IPFIX flexible. [3, 82] Table 1 below shows an example of simple information sent via IPFIX.

Table 1: IPFIX Information

Source	Destination	Packets
192.168.2.2 192.168.2.3	192.168.1.1 192.168.1.4	235 48

The information set would be sent in a message that looks like the message shown in table 2 below.

Table 2: IPFIX message. Reprinted from Ding (2010). [3, 84]

Bits 0……..15	Bits 16……31
Version= 0x000a	Message Length = 64 Byte
Export Timestamp = 2014-1-23 15:59:04	
Sequence Number = 0	
Source ID = 12345678	
Set ID =2 (Template)	Set Length = 20 Bytes
Template ID = 256	Number of fields =3
Typ = Source Ipv4 Address	Field length = 4byte
Typ= Destination Ipv4Address	Field length = 4byte
Typ= Packet Delta Count	Field length = 8byte
Set ID =256 (Data Set using Template 256)	Set Length = 24 Byte
Record 1, Field 1 = 192.168.2.2	
Record 1, Field 2 = 192.168.1.1	
Record 1, Field 3 = 235 packets	
Record 2, Field 1 = 192.168.2.3	
Record 2, Field 2 = 192.168.1.4	
Record 3, Field 3 = 48 Packets	

As seen in table 2 above, the message has an IPFIX header and two sets: One template set that introduces the built-up of the data set used and also the data set that contains the real data information. The template set will then be buffered in the collectors and hence it will not be transmitted in the subsequent message. [3, 84]

3 Methodology

The IPFIX analysis of this project was carried out in a networking environment. It required designing a network that depicts a real-life network that generates the required traffic for analysis.

3.1 Materials

A network was built using three Cisco routers, a Cisco switch, an Internet port and three computers that were used to partly configure the Cisco devices and also to act as the collectors of the traffic. The Cisco routers used were Cisco 2800 series module with an IOS image C2800NM-ADVENTERPRISEK9-M Version 15.1(4)M6. This type of router can accommodate NetFlow configuration.

The Cisco switch used was a layer 3 switch Cisco catalyst 3560 series to help with the distribution of the internal network. Three Dell Optiplex 755 computers were used to collect the traffic and also host the software. The computers were 64-bit operating systems, with an installed memory of 4 gigabytes. They are the Intel core with due central processing unit (CPU) with a 2.66 GHz speed. The hard drive in the computers was at least 240 gigabytes in order to be able to collect as much information as possible. These were strong enough machines that could be used to run the software being used in these experiments.

3.2 Network Design

The Router Open Shortest Path First (OSPF) was used while designing the network. This was because I am more comfortable with this routing protocol. As seen in figure 6 below three routers where connected with a serial link connecting each router to the other with one in the middle. R2 connecting the others R1 and R3. The serial link gave another dimension of collecting flows from different port types.

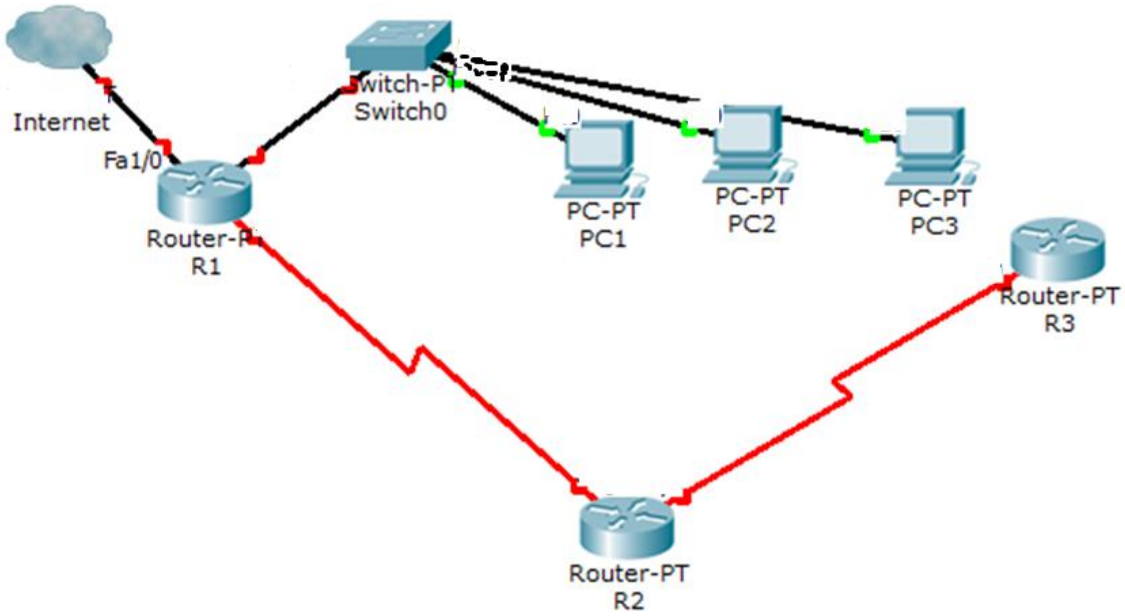


Figure 6: Network design

The routers were connected with serials cables. The link between R1 and R2, The DCE is at the Serial port s0/0/0 at R1 and the link between R2 and R3, the DCE link was at R2 S0/0/1 as shown above. The first Ethernet port f0/1 on R1 was connected to an internet port while F0/0 was connected to a Cisco switch that was connected to the computers or the collectors. There was full connection between all the devices from the computers to the routers with OSPF routing protocol used. NetFlow was enabled on the three routers and aggregated flows were collected at each computer.

The traffic was generated with the pings from the computers and the devices and since the network was connected to the internet, HTTP traffic was generated as well.

3.3 Software

A shortlist of software was studied, and according to the resources available, three types of software were chosen. This was based on how much bytes they had and the operating system on which it should run. Since I have limited knowledge of Linux, I chose only Windows based software. I also chose free or partially free software.

Three computers hosted the different types of software each, so as to analyze every type of software independently and also because of the space requirement for each software, they were installed one per computer.

3.3.1 Manage Engine NetFlow Analyzer

The NetFlow analyzer is a complete traffic analytic tool. It stores the forwarded traffic in a network, analyses it, and displays real-time visibility of the network design and performance such as bandwidth usage. It also helps in security analysis and gives out an easy-to- read web page display of the entire network. [13] It has a free trial version for 30 days and a limited free edition that limits the number of interfaces to monitor.

The system requirements are shown in table 3 below.

Table 3: System requirements. Reprinted from Manage Engine (2014). [13]

Flow Rate (in-flows / sec-ond)	Processor	RAM	Hard-disk space	server type
0 – 3000	2.4 GHz Dual core	2GB	250 GB	-
3000 – 6000	3.2 GHz Quad Core	4GB	600 GB	-
6000 – 9000	3.2 GHz Quad Core	8GB	1 TB	64-bit
above 9000	3.2 GHz Quad Core	8GB	1TB	64-bit

As shown in table 3 above, the more in-flow expected in the system, the more strongly the machine that is required to analyze the traffic. NetFlow communicates using the following ports that must be left unblocked or not to be assigned of any other commu-nication:

- Web server port 8080 for TCP connection to the web browser.
- NetFlow listener port 9996, UDP port for receiving traffic from the routers.

- MySQL port 13310, to connect to the MySQL database in NetFlow Analyzer.
- SNMP port 161, UDP, to poll the device and forwarding the devices' specifications.

3.3.2 PRTG Network Monitor

The PRTG is a powerful network monitoring application for Windows-based systems. It has a free unlimited trail version for only 30 days and the free edition is limited to a few number of sensors. It is efficient to small, medium and large networks as it is capable of monitoring LAN, WAN, WLAN and VPN. It is also capable of monitoring both physical and virtual webs, mail, and file servers. It monitors several network parameters including network availability, bandwidth consumption, Quality of Services, CPU usage, and memory load. It displays a live reading on the web page of the entire network system to the engineer operating the system. [14, 14]

The PRTG is very easy to install and also to use and it uses the Simple Network Management Protocol (SNMP), Windows Management Instrumentation (WMI) for monitoring windows-based systems, packet sniffer, Cisco NetFlow, IPFIX and JFlows. The PRTG records network parameters and stores the data in its local data base for analysis later. [14, 14]

PRTG has the following system requirements that need to be met before installing the software:

Supported Operating systems

- Microsoft Windows XP SP2 or later
- Microsoft Windows Server 2003 SP1 or later
- Microsoft Windows Vista
- Microsoft Windows Server 2008
- Microsoft Windows Server 2008 R2
- Microsoft Windows 7
- Microsoft Windows 8
- Microsoft Windows Server 2012**
- Microsoft Windows Server 2012 R2 ** [14, 20]

As seen in the list above, PRTG can be installed on Windows based operating systems starting with Windows XP.

The hardware requirements for PRTG are a system with a Central Processing Unit built at latest 2007which can monitor 1000 sensors with RAM memory above 1024 MB and a hard disk drive of capacity 200kB of disk space per sensor per day. [14, 22] PRTG requires asynchronous java Script and XML to display the output on a browser, the browsers with these requirements are Google Chrome 31 or later, Mozilla Firefox 25 or later and Microsoft Internet Explorer 10 or 11. [14, 22]

The devices to be monitored should be equipped with SNMP, WMI or xflow (IPFIX, NetFlow) so that they can communicate with the software and send the required packets to be analyzed. Only data packets that pass the local machine's network card can be analyzed using sniffing. [14, 24]

3.3.3 SolarWinds Network Performance Monitor

The SolarWinds Network performance monitor (NPM) controls fault and network performance management that scales with rapid network growth and expands with the network monitoring needs, allowing collecting and viewing data in real time and also to analyse historical statistics directly from the web page. [15, 23] SolarWinds NPM has a free full trial of one month and a limited free edition to the number of devices that can be monitored. It gives out a straight forward web display of the network and its events and it is also easy to install. SolarWinds can be used to monitor the network availability, bandwidth capacity utilization and buffer usage and errors for both physical and virtual devices of the network. [15, 18]

SolarWinds NPM provides graphs, tables and lists on to the web page and has sensors for which one can set to monitor a specific area on interest. SolarWinds uses the following networking terminologies to collect packets from monitored devices:

- Internet Control Message Protocol (ICMP)
- Simple Network Management Protocol (SNMP)
- Management Information Base (MIB)
- Windows Management Instrumentation (WMI) [15, 28]

These terminologies help in the transportation of the packets from the managed devices to the collectors and helping in authentication in order to control security in the network management system.

4 Results

After setting up the network running and all the programs installed on different computers, small traffic was generated in the network and the program's web consoles were opened on the different computers. Cisco can only send traffic to only two destinations, so I had to use two programs at a time at most.

4.1 Network Discovery

The devices to be monitored can be either manually added to the agent system or they can be discovered automatically with the auto discovery application on all the three types of software.

Auto discovery is done in three steps. First the program will scan the network with a ping. This means there must be a connection between the devices and the computer hosting the program. Secondly the program will use SNMP to identify the types and credentials of the devices. This will include the device type, name and model, to mention a few. For PRTG sensors will be created and they should match the discovered devices based on the templates built in the devices discovered. These include the ping sensor, CPU load sensors and temperature sensors.

Figure 7 below shows the discovered network of this project. All the devices where discovered automatically with all the programs.

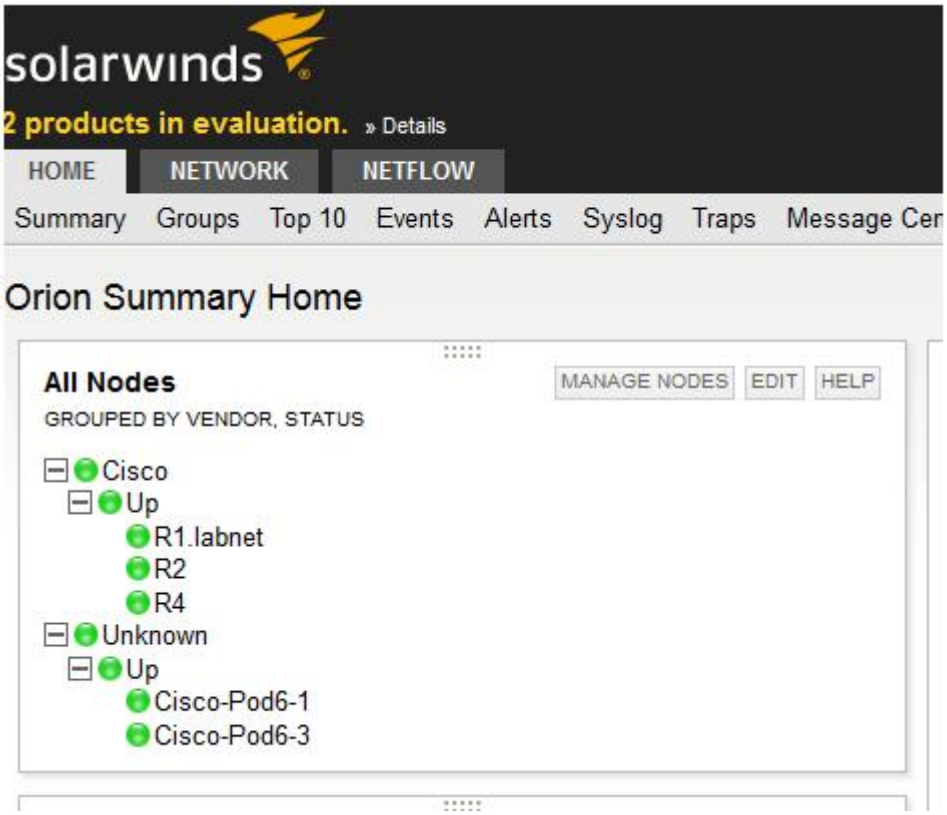


Figure 7: Network discovery with SolarWind

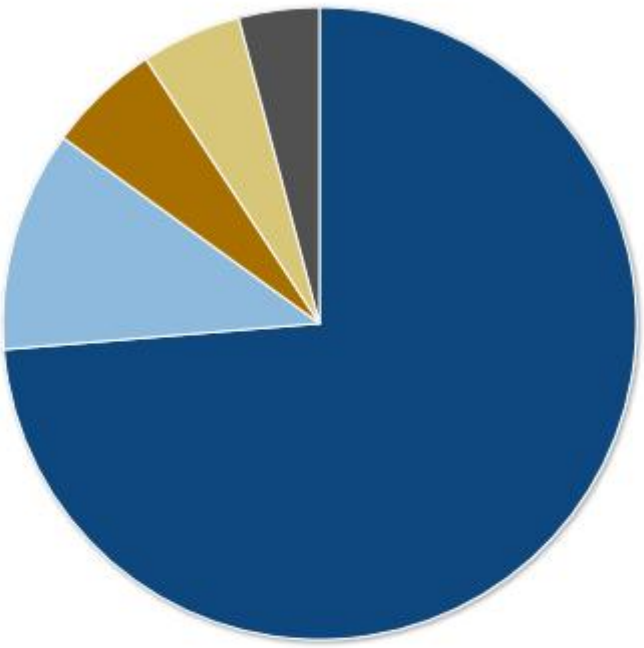
Since the network was all running, all the nodes appeared green, showing that there was no problem with the network as shown in figure 7 above. The PRTG and Manage Engine also displayed a ‘green’ network implying that the network had no problems.

4.2 Resource Usage

With a little traffic flowing through the network, the network resorce utilization would be easily viewed by the graphs and pie chats that were displayed on the web page of the different software. SolarWind NetFlow analyzer showed the Top 5 Applications in the network, as shown in figure 8 below.

Top 5 Applications EDIT HELP

BOTH, LAST 1 HOURS



APPLICATION	INGRESS BYTES	EGRESS BYTES	INGRESS PACKETS	EGRESS PACKETS	PERCENT
World Wide Web HTTP (80)	11.6 Mbytes	11.6 Mbytes	10.55 k	10.54 k	73.32%
SNMP (161)	1.9 Mbytes	1.6 Mbytes	18.56 k	14.7 k	11.24%
RTP (UDP)	907.0 kbytes	907.0 kbytes	2.44 k	2.44 k	5.75%
http protocol over TLS/SSL (443)	806.2 kbytes	799.7 kbytes	2.28 k	2.25 k	5.09%
Unmonitored traffic	646.1 kbytes	645.8 kbytes	1.14 k	1.13 k	4.1%
Remaining traffic	150.3 kbytes	5.9 kbytes	840	126	0.5%

Figure 8: Top 5 applications from SolarWind

The World Wide Web HTTP takes the biggest bandwidth as shown in figure 8. It takes 72% of the bandwidth of the network. This is followed by the SNMP communications that takes 11% and then the UDP follows with 5.7%. SolarWind and PRTG display the data in pie chats and time stamped log messages. The Manage Engine shows the date in graphs that can be easily read, and the moving of a cursor in the graph will display the data of that particular point. The programs also displayed more detailed information for every device. By analyzing a particular router (in this case I concentrated more on router R1 with a link to the internet), its ports availability or unavailability can be seen with the colors that every node displays, green representing good conditions, yellow worning and red critical condition (down or over utilized). A

grey color means that one port on a router was having a problem but the rest were working fine.

Traffic can also be analyzed for every application. Expanding/ clicking the plus sign on the left of the World Wide Web HTTP. Routers that have that application flow are shown. In this case only R1 is shown. When R1 is expanded, it will show the interfaces that have HTTP traffic flowing through them. In this case only the FastEthernet ports have that traffic flowing through them.

Considering FastEthernet 1 by clicking it, the graghs of the date are displayed. Figure 9 below shows the top five traffic sources by country using SolarWind

COUNTRY	INGRESS BYTES	EGRESS BYTES	INGRESS PACKETS	EGRESS PACKETS	PERCENT (UTILIZATION)
United States	0 bytes	7.9 Mbytes	0	6.47 k	74.01% (0.02%)
Europe	0 bytes	1.0 Mbytes	0	896	9.72% (0%)
Finland	0 bytes	935.0 kbytes	0	1 k	8.72% (0%)
Sweden	0 bytes	363.2 kbytes	0	342	3.39% (0%)
Private Address	357.2 kbytes	0 bytes	3.7 k	0	3.33% (0%)
Remaining traffic	0 bytes	88.5 kbytes	0	86	0.83% (0%)

Figure 9: Top 5 traffic sources by country SolarWinds

Figure 9 shows that the source of the traffic in the network is from the United States with the percentage of 74. Europe is second with 9.7% , Finland with 8.7%, Sweden

with 3.4% and the private addresses in the network being source to the traffic with 3.3%.

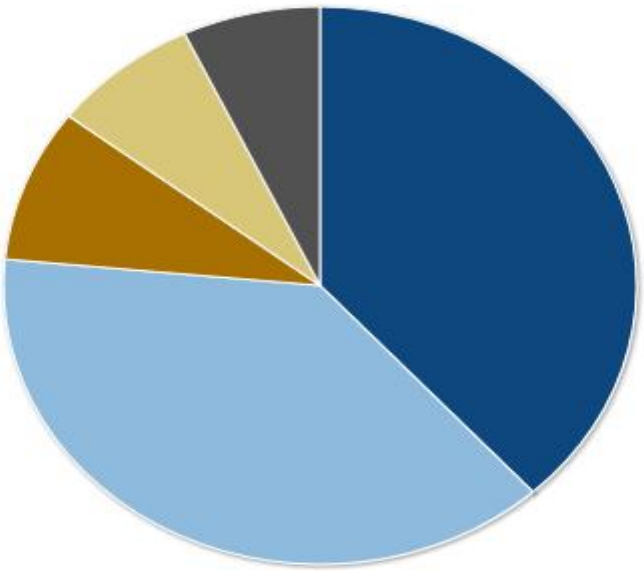
There are also other graphs that are displayed from this interface. They include the type of service, the top 5 transmitter addresses, the top 5 protocols of the traffic transfer where TCP is 100%, the Top 5 receivers , total packets transferred and top 5 conversations.

4.3 Conversations

SolarWind and PRTG also display the top conversation links where they display the connections between source IP addresses and destination IP addresses in a pie chat and a message box below the chat. In the message box, there is added information such as the bytes used in the conversations and the percentage of the bandwidth usage. This is shown in figure 10 below.

Top 5 Conversations EDIT HELP

BOTH, LAST 1 HOURS



CONVERSATION	INGRESS BYTES	EGRESS BYTES	INGRESS PACKETS	EGRESS PACKETS	PERCENT
Between 10.94.62.126 and 109.105.109.207	4.7 Mbytes	4.7 Mbytes	3.66 k	3.66 k	29.05%
Between 109.105.109.207 and Cisco-Pod6-1 (192.168.2.4)	4.7 Mbytes	4.7 Mbytes	3.66 k	3.66 k	29.05%
Between 10.0.0.2 and Cisco-Pod6-1 (192.168.2.4)	1.2 Mbytes	1.0 Mbytes	5.28 k	4.44 k	6.75%
Between 10.0.3.2 and Cisco-Pod6-3 (192.168.2.3)	958.4 kbytes	897.8 kbytes	6.47 k	5.82 k	5.75%
Between 10.0.0.2 and Cisco-Pod6-3 (192.168.2.3)	894.0 kbytes	833.3 kbytes	3.75 k	3.1 k	5.35%
Remaining traffic	4.2 Mbytes	3.6 Mbytes	16.02 k	11.59 k	24.05%

Figure 10: Top 5 conversations from SolarWinds.

Almost 60 % of the resources is consumed in talks between source 10.94.62.126 and an external address destination 109.105.109.207 and between PC 1 and 109.105.109.207, as shown in figure 10 above

In the case of shortage of bandwidth or overload, the engineer can easily find out which source is consuming more resources and which sites the one using the biggest bandwidth is visiting. Here he or she can either block those sites or adds more resources to the network.

4.4 Tests

I created two tests in the network so that I could analyze how the software would alert with the different tests, I selected only two management tests that I concentrated on which were fault management and bandwidth usage.

4.4.1 Fault Management

With the network set running with a little traffic being generated (Ping), the network is broken by shutting up an interface, hence stopping the flow of traffic from the diverce with a shutdown interface and the rest of the network. The interface shutdown is the serial on router 1 connecting to router 2.

Since the serial on R1 connects the other routers to the traffic collectors, first a warning on R1 will be displayed in the network view window on the web page. At this instance the collectors will not have any communication with router 2 and router 4 in the network, so they are labelled as off or red, as shown in figure 11 below.

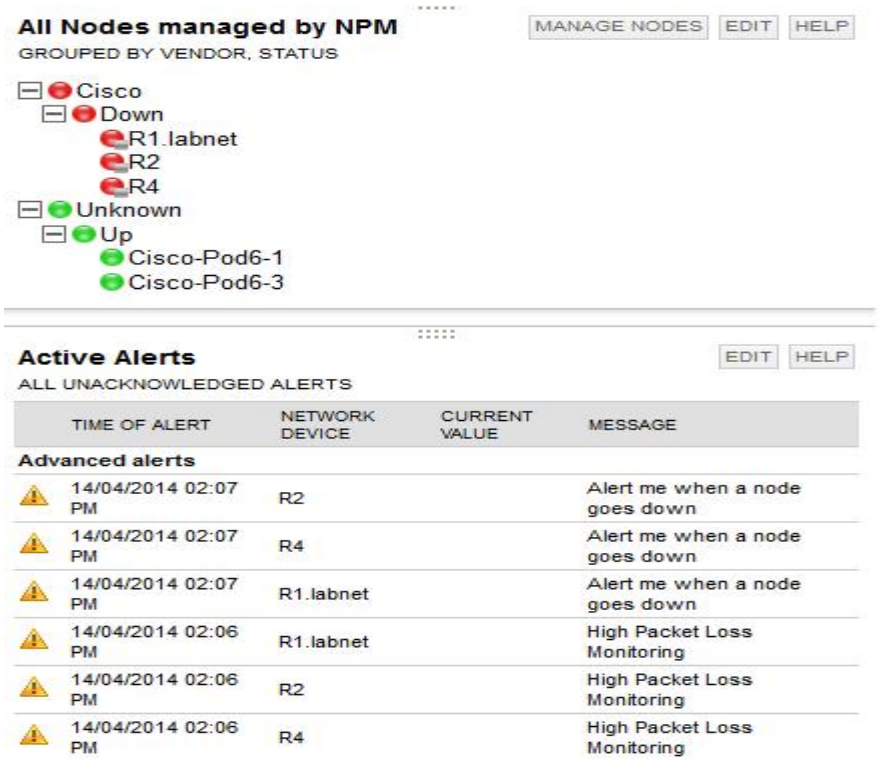


Figure 11: Fault display with SolarWind

Figure 11 shows that all the network turns red because of the connection lost in the network. There is no more traffic collection at the collectors. The alert messeges below in the SolarWind program specify the problem as " Alert me when a node goes down" on R1 and R2 and "High Packet loss Monitoring" on R2 and R4.

The PRTG displayed red colour to the PING sensors for all the routers that cannot be pinged. Since ping is the first step in the network discovery, as mentioned in section 4.1, all the other sensors are paused with the blue color as shown in figure 12 below.

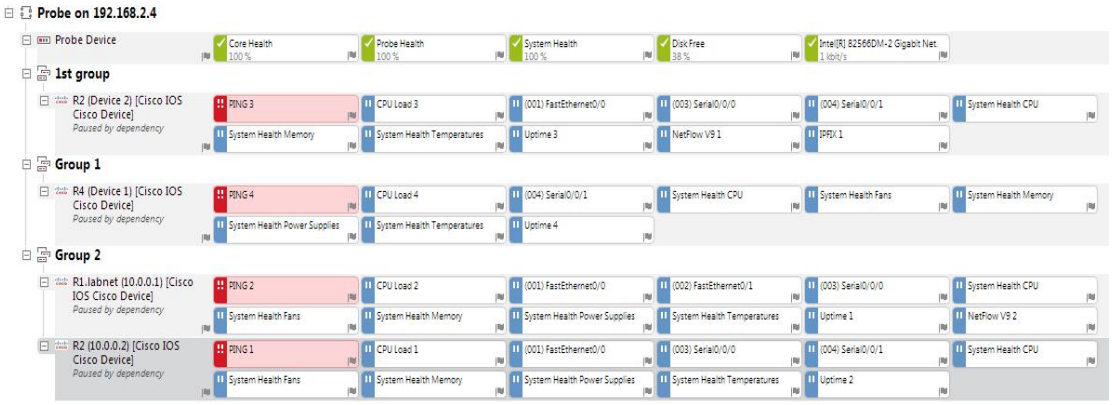


Figure 12: Fault display with PRTG

Figure 12 does not show the exact problem in the network but with network knowledge, the ping can specify that it is a connection problem because its only the ping sensors that are red. Also the logged messages show that the first error message is displayed on router R1 displaying a message 'no response from interface s0/0/0'.

4.4.2 Bandwidth Management

A situation was created where the bandwidth was stretched in the network. This was done by watching a video from Youtube on PC 1 in the network and analyzing what was being displayed on the monitoring web pages. First, notifications were set up on the PRTG network monitor to alert with the red color when the bandwidth of any interface on the divece exceeded 80% of the total bandwidth. An email had to be sent to the operator during the time between 08:00 and 18:00 on working days and only if the same condition continued for at least 300 seconds in the non working time periods. The email should be sent only once.

The PRTG can notify the operator in many ways, for example by sending emails, sending syslogs or sending SMS messages. I used the email because it was free and suited the working environment. The message in the email in the notification described the device name which had the problem, the time when the problem occurred and the reason, with a lot other information. This email message was changed to be more clear.

Viewing the network summary, there was a warning grey colour sign on router R1 and the problem was seen on both the FastEthernet ports that were passing more traffic than the set 80% of the total bandwidth. Navigating to the Fastethernet port 01, sparks in the graphs could be seen on the web page. Figure 13 below shows the top 5 protocols with the horizontal exes displaying the time, while the vertical represents the number of bytes of the packets.

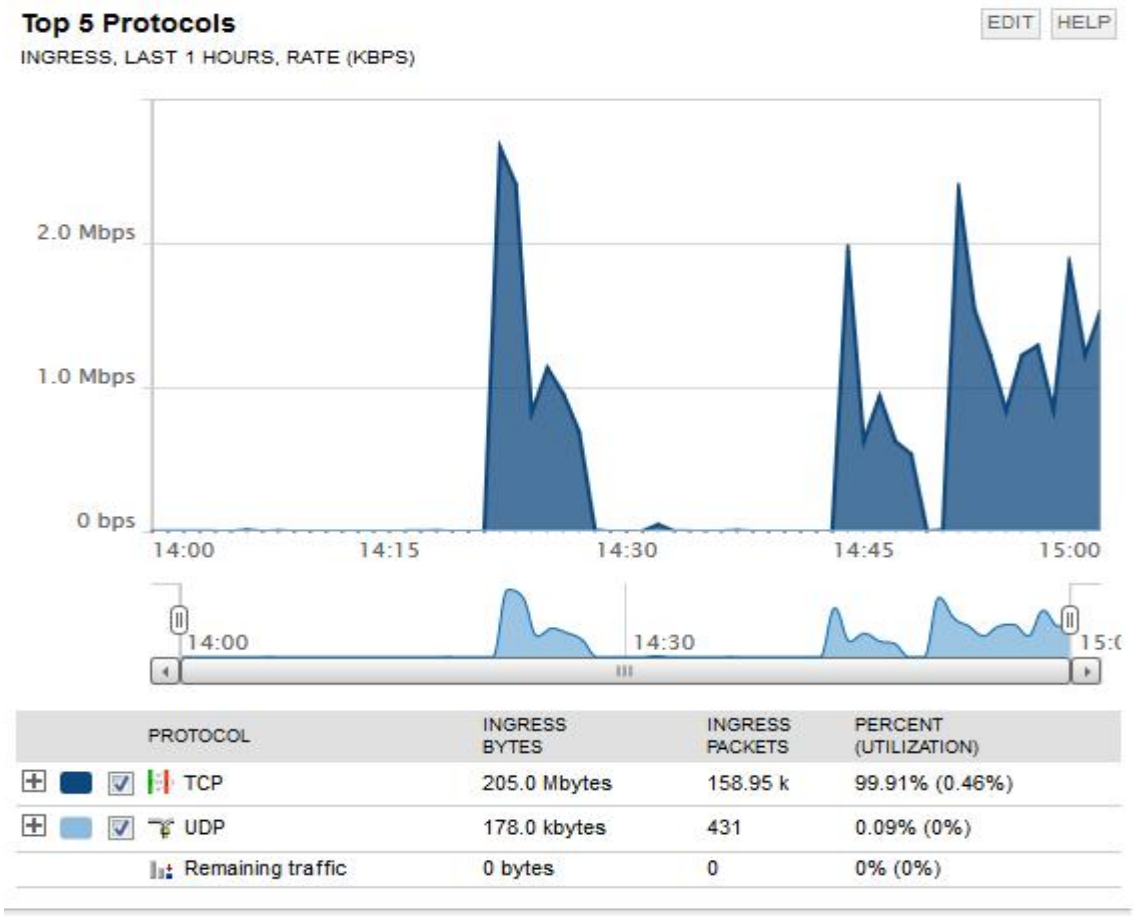


Figure 13: Top protocols on interface F0/1 from SolorWinds

Almost all the traffic is using the TCP protocol. The volume of incoming traffic is 205 Mbytes as seen in figure 13, with 99.9% of the total traffic flowing through that interface. The maximum speed of the interface can be seen when the cursor was moved to the peak in the graph. The reading was 2.4 Mbps and the date was also shown (Tuesday April 15, 2014 14:51).

The top 5 end points are shown in another graph with IP addresses of the end points of the communications with the number of packets transferred. When an IP address was clicked, the details of that address were given. It was not possible to look up the address because of the Domain Name Server but the country of the host IP address was shown and the total traffic transmitted was shown in figure 14.

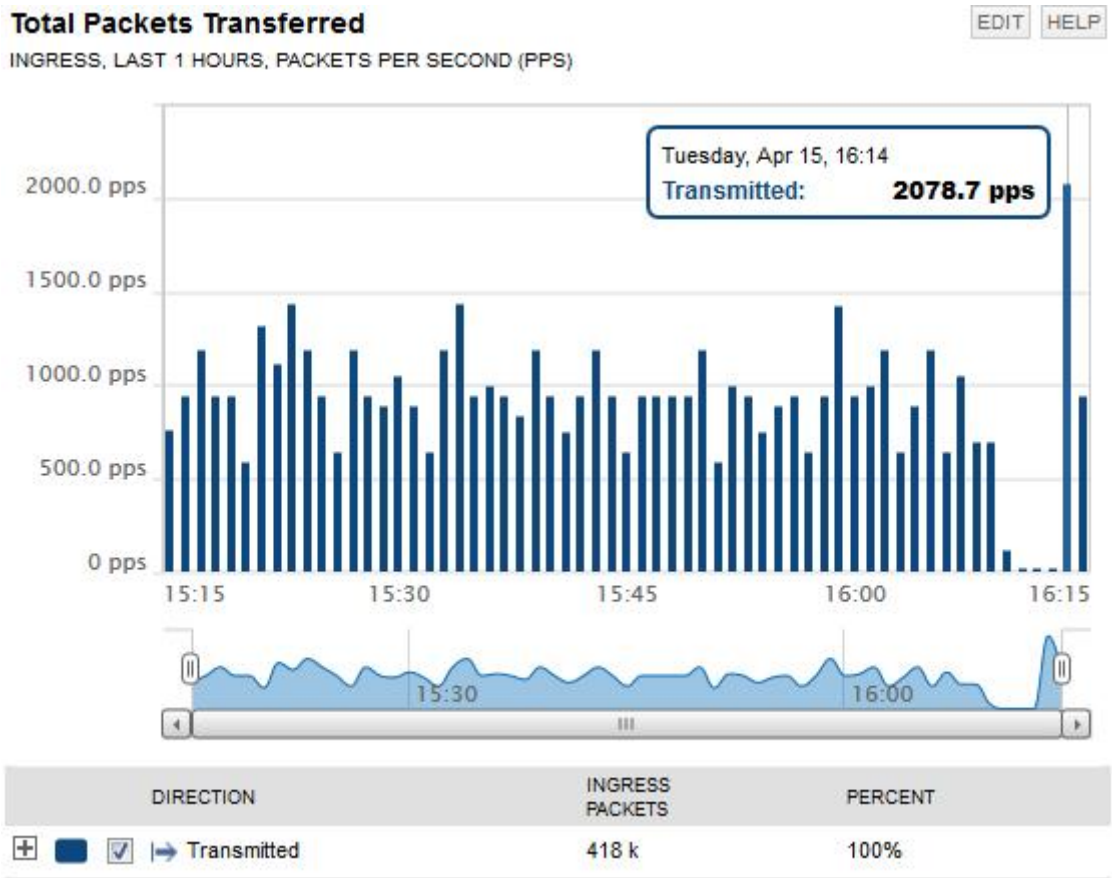


Figure 14: Total packets to address 173.194.48.78 (top end point)

The packet count in figure 16 above is measured per second and the highest packet transfer was seen at 16:14 and the packet size was 2078.7 packets per second. This happened when another Youtube video was played.

5 Discussion

The three types software give a web page control system which is easy to both read and analyse. With the easy time graphs, the instructor monitoring the system can notice the history of the traffic flow at a given time. It is easy to notice the overused bandwidth with the peaks in the graphs at a particular time. The pie charts give the overall comparison between different elements such as protocols. With these graphs and charts, the instructor monitoring the network system can either increase the capacity of the network or block any port or address as he or she requires to keep the system at the required service level.

In this project, because of the local and private networks, it was impossible to translate all the addresses as I had thought would come out. More research is needed to identify the private addresses even after the network address translation. The practical part of the project was carried out in the Cisco Lab at the Helsinki Metropolia University of Applied Sciences with the lab devices. Being the only one using these devices for four months has given me a useful opportunity to try out networking projects with out intervention from other students.

Although the types of software were either free or partially free, for the partially free software, I had to un install the software every month, register again under different names and download the trail full version because I wanted to have a fully unlimited operating monitoring system. This was sometimes hard because I had to lose all the results from the analysis of the collected packets I had achieved throughout the previous month.

With IPFIX, not only have I learnt to monitor traffic flowing in a network but also analyze it. I also learnt about the SNMP and the NetFlows in detail. I have a task to start learning Linux as many types of networking software are Linux-bassed and free and yet my knowledge of Linux is limited

6 Conclusion

The goal of the project was to first look for the software that could suit both my knowledge and be either free or partially free and could run on the available resources on the Networking lab computers. Three different software were selected from a list of software that could support IPFIX. These included the PRTG Network Monitor, SolarWind NetFlow Performance Monitor and Manage Engine NetFlow Analyzer. The second goal was to create traffic in the network and analyze it with the three different software using IPFIX. Traffic was generated and the flows were analyzed on the different software. The software could easily show the source addresses and destination addresses of the traffic, the source and destination countries, the bandwidth usage on the devices and every specific port, the protocol of communication used for every traffic and also the 'high talkers' in the network. With the available data on the graphs and pie chats, the network manager can easily see what is going on in the network, where to assign more bandwidth and which ports to close in a managed network.

With the increased network global interlinking, the network usage is increasing on an every day basis. This has increased the number of Internet service providers in a profitable business all over the world. The main competetive task is to provide a reliable (always available) quality service but administartors cannot garentee these services unless they investigate what exactly their customers are using the network for. The administrators should protect their customers from threats and viruses that could come from hackers or unauthorized sites. Hence the administrators are required to carry out network management to monitor all the traffic in and out of the network.

Network management and the collection of data about the flow of traffic in networks has taken place for some time now with Cisco using the Simple Network Management Protocol. Cisco has also introduced NetFlows to collect the traffic flow in the networks but still something has been missing. The IPFIX was introduced based on NetFlow V9 to standardize the collection of traffic in the network from the devices to a particular place (collector) with a characteristic feature of distinguising different flows. Packets with the same source address, destination address, source and destination port and the same layer 3 protocol are grouped to be the same flow.

References

1. Cisco. Introduction to Cisco IOS NetFlow - A Technical Overview. San Jose, CA; May 2012. URL: http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod_white_paper0900aecd80406232.html. Accessed 30 March 2014.

2. Kevin Dooley , Ian J. Brown. Cisco Cookbook. O'Reilly Media; June 2003. URL: http://f3.tiera.ru/1/SNMP_COOK.pdf Acceseed March 20 2014

3. Jianguo Ding. Advances In Network Management. Broken NW: Auerbach Publications; 2010.

4. The availability Digest: High Availability Network Fundamentals. April 2009. Sombers Associates, Inc and W.H Highleyman URL: http://www.availabilitydigest.com/public_articles/0404/ha_networks.pdf. Accessed April 22 2014.

5. Matti Puska, M. Sc, Principal Lecturer. Network Management In IP Networks. Espoo: Metropolia University of Applied Science; 2014.

6. Cisco. Network Management System: Best Practices White Paper. San Jose, CA; June 2007. URL: http://www.cisco.com/c/en/us/support/docs/availability/high-availability/15114-NMS-bestpractice.html. Accessed March 28 2014.

7. Cisco. Configuration Management: Best Practices White Paper. San Jose CA; November 2006. URL: http://www.cisco.com/c/en/us/support/docs/availability/high-availability/15111-configmgmt.html. Accessed April 3 2014.

8. Diana Teare. Structuring and Modularizing the Network with Cisco Enterprise Architecture. Indianapolis Indiana: Pearson Education, Cisco Press. June 2008. URL:http://www.ciscopress.com/articles/article.asp?p=1073230&seqNum=4. Accessed April 9 2014.

9. Waldbusser S, Cole R. Introduction to the Remote Monitoring (RMON) Family of MIB Modules. August 2003. URL: https://tools.ietf.org/html/rfc3577. Accessed April 4 2014.

10. Javvin Technologies Inc. Network Protocols handbook, second edition. Saratoga USA; 2005.

11. Caligare S R O. NetFlow Export Format. Prague Czech Repulic; May 10 2006. URL: http://netflow.caligare.com/netflow_format.htm. Accessed 1 April 2014.

12. Ye Tung, Hao Che. A flow caching mechanism for Fast Packet Forwarding. Pennsylvania USA: Pennsylvania state University. November 2001. URL: http://crystal.uta.edu/~hche/PUBLICATIONS/papers/flow-caching-computer-communications-proof-version.pdf. Accessed March 30 2014.

13. ManageEngine. ManageEngine ServiceDesk Plus Admin Guide. USA; 2010. URL: http://www.manageengine.com/products/servicedesk/help/ManageEngine_ServiceDesk Plus_8_Help_AdminGuide.pdf. Accessed 4 April 2014.

14. Paessler. PRTG Network Monitor User Manual. Nuremberg; April 2014. URL: http://download-cdn.paessler.com/download/prtgmanual.pdf. Accessed 8 April 2014.

15. Solarwinds Orion. Network Performance Monitor Administrator Guide. Oakland California: ComponentOne; December 2011. URL: http://www.solarwinds.com /documentation/Orion/docs/OrionAdministratorGuide.pdf. Accessed April 9 2014.

Configuration for Router 1

```
R1>en
R1#sh run
Building configuration...
Current configuration : 2062 bytes
version 15.1
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
hostname R1
boot-start-marker
boot-end-marker
no aaa new-model
!
memory-size iomem 5
clock timezone 0 0 0
!
dot11 syslog
ip source-route
!
ip cef
!
ip dhcp excluded-address 192.168.2.1 192.168.2.5
!
ip dhcp pool MUSA
 import all
 network 192.168.2.0 255.255.255.0
 default-router 192.168.2.1
 dns-server 10.94.1.4
ip flow-cache timeout active 1
no ipv6 cef
!
multilink bundle-name authenticated
```

```
!
voice-card 0
!
crypto pki token default removal timeout 0
license udi pid CISCO2811 sn FCZ133770S6
!
redundancy
!
interface Loopback0
 ip address 3.3.3.3 255.255.255.0
 ip ospf network point-to-point
!
interface FastEthernet0/0
 ip address 192.168.2.1 255.255.255.0
 ip flow ingress
 ip flow egress
 ip nat inside
 ip virtual-reassembly in
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address dhcp
 ip flow ingress
 ip flow egress
 ip nat outside
 ip virtual-reassembly in
 duplex auto
 speed auto
interface Serial0/0/0
 ip address 10.0.0.1 255.255.255.0
 ip flow ingress
 ip flow egress
 ip nat inside
```

```
 ip virtual-reassembly in
 clock rate 64000
!
interface Serial0/0/1
 no ip address
 shutdown
 clock rate 2000000
router ospf 1
 network 3.3.3.0 0.0.0.255 area 0
 network 10.0.0.0 0.0.0.255 area 0
 network 192.168.2.0 0.0.0.255 area 0
 default-information originate
ip forward-protocol nd
no ip http server
no ip http secure-server
ip flow-export version 9
ip flow-export destination 192.168.2.3 9996
ip flow-export destination 192.168.2.4 9996
ip nat inside source list 101 interface FastEthernet0/1 overload
ip route 0.0.0.0 0.0.0.0 10.64.62.254
access-list 101 permit ip 192.168.2.0 0.0.0.255 any
access-list 101 permit ip 10.0.0.0 0.255.255.255 any
snmp-server community public RO
control-plane
mgcp profile default
line con 0
line aux 0
line vty 0 4
 login
 transport input all
scheduler allocate 20000 1000
end
```

Configuration for Router 2

```
2#sh run
Building configuration...


Current configuration : 1600 bytes
!
version 15.1
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R2
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
memory-size iomem 5
clock timezone 0 0 0
!
dot11 syslog
ip source-route
!
!
ip cef
!

ip flow-cache timeout active 1
no ipv6 cef
!
multilink bundle-name authenticated
```

```
voice-card 0
!
crypto pki token default removal timeout 0
!
license udi pid CISCO2811 sn FCZ133770S0
!
redundancy
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 ip flow ingress
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/0/0
 ip address 10.0.0.2 255.255.255.0
 ip flow ingress
!
interface Serial0/0/1
 ip address 10.1.0.1 255.255.255.0
 ip flow ingress
 clock rate 64000
!
router ospf 1
 network 10.0.0.0 0.0.0.255 area 0
 network 10.1.0.0 0.0.0.255 area 2
 network 192.168.1.0 0.0.0.255 area 0
 default-information originate
!
```

```
ip forward-protocol nd
no ip http server
no ip http secure-server
!
ip flow-aggregation cache protocol-port
 cache timeout active 1
 export version 9
 export destination 192.168.1.2 9996
 enabled
!
ip flow-aggregation cache source-prefix
 cache timeout active 1
 export version 9
 export destination 192.168.1.2 9996
 enabled
!

snmp-server community public RW
snmp-server ifindex persist
!
control-plane

mgcp profile default
line con 0
line aux 0
line vty 0 4
 login
 transport input all
scheduler allocate 20000 1000
end
```

Configurations for router 4

```
R4#sh run
Building configuration...


Current configuration : 1383 bytes
!
version 15.1
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R4
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
memory-size iomem 5
!
dot11 syslog
ip source-route
!
!
ip cef
!
ip flow-cache timeout active 1
no ipv6 cef
!
multilink bundle-name authenticated
!
```

```
voice-card 0
!
crypto pki token default removal timeout 0
!
license udi pid CISCO2811 sn FCZ122272L9
!
redundancy
!
interface Loopback0
 ip address 5.5.5.5 255.255.255.0
 ip ospf network point-to-point
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex auto

speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/0/1
 ip address 10.1.0.2 255.255.255.0
 ip flow ingress
```

```
 ip flow egress
!
router ospf 1
 network 5.5.5.0 0.0.0.255 area 2
 network 10.1.0.0 0.0.0.255 area 2
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
ip flow-export version 9
ip flow-export destination 192.168.2.3 9996
ip flow-export destination 192.168.2.4 9996
!
snmp-server community public RO
control-plane
mgcp profile default
!
line con 0
line aux 0
line vty 0 4
 login
 transport input all
!
scheduler allocate 20000 1000
end

R4#
```