UltraFlow
-Cisco Netflow tools-

UltraFlow

UltraFlow is an application for collecting and analysing Cisco Netflow data. It is written in Python, wxPython, Matplotlib, SQLite and the Python based Twisted network programming framework.

UltraFlow is designed with the following following objectives:-

- Intuitive Interface
- Simple, fast and easy to use
- Well Documented
- up-to-date reports and graphs
- Powerful analysis features
- Attractive statistics
- Database driven
- Easy installation and configuration
- Easily modifiable to meet a specific organisation's network environment

Cisco Netflow

NetFlow is a Cisco developed network protocol that runs on Cisco IOS-enabled equipment like routers and switches. It monitors and records all traffic passing through the supported NetFlow device. It is used for collecting IP traffic information.

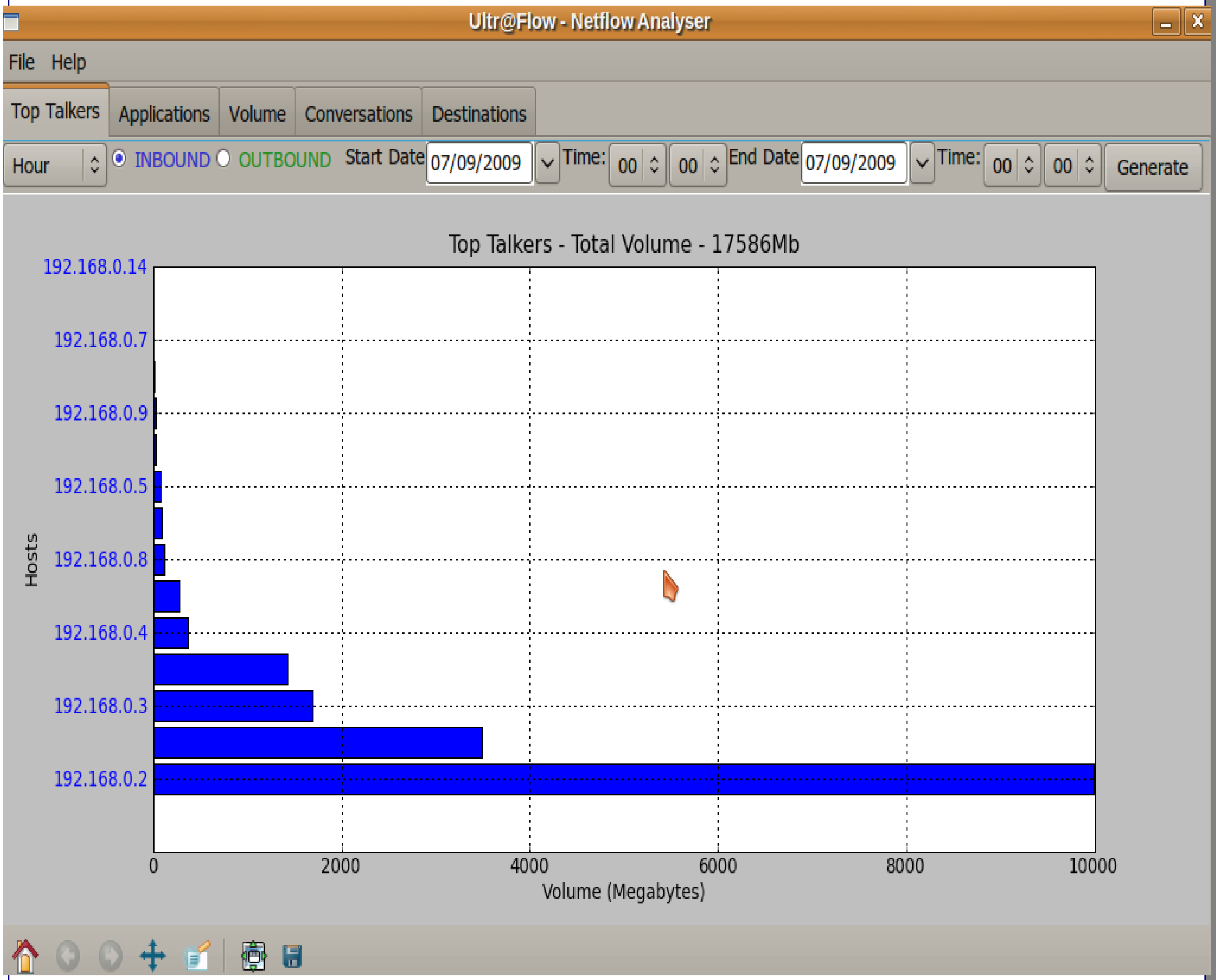
Here are some of the uses of Cisco Netflow:-

- Network bandwidth monitoring without using using expensive hardware probes
- Billing and accounting. Organisations can use NetFlow to track IP traffic flowing into or out of their networks for to implement usage-based billing
- Network planning and capacity planning. NetFlow data can be captured over a long period of time which enables users to track and anticipate network growth and plan upgrades to increase the number of devices, ports, or high bandwidth interfaces.
- Combating denial of service (DoS) attacks, security, detection of unauthorised WAN traffic and anomaly detection. Changes in network behaviour indicate anomalies that are clearly demonstrated in NetFlow data
- Comparing Application Usage Patterns and understanding impact of new applications introduced to the network
- Validating QoS Implementations
- identifying underutilised links
- Generating usage trends and help in capacity planning

UltraFlow Features

Top talkers

These reports show you which hosts are using the most bandwidth. You can select specific time period like last hour, week, month, year or specific date and time. From the top talkers report further investigation can be done using the *conversations, applications and destinations* sections of the application to determine the traffic movement.

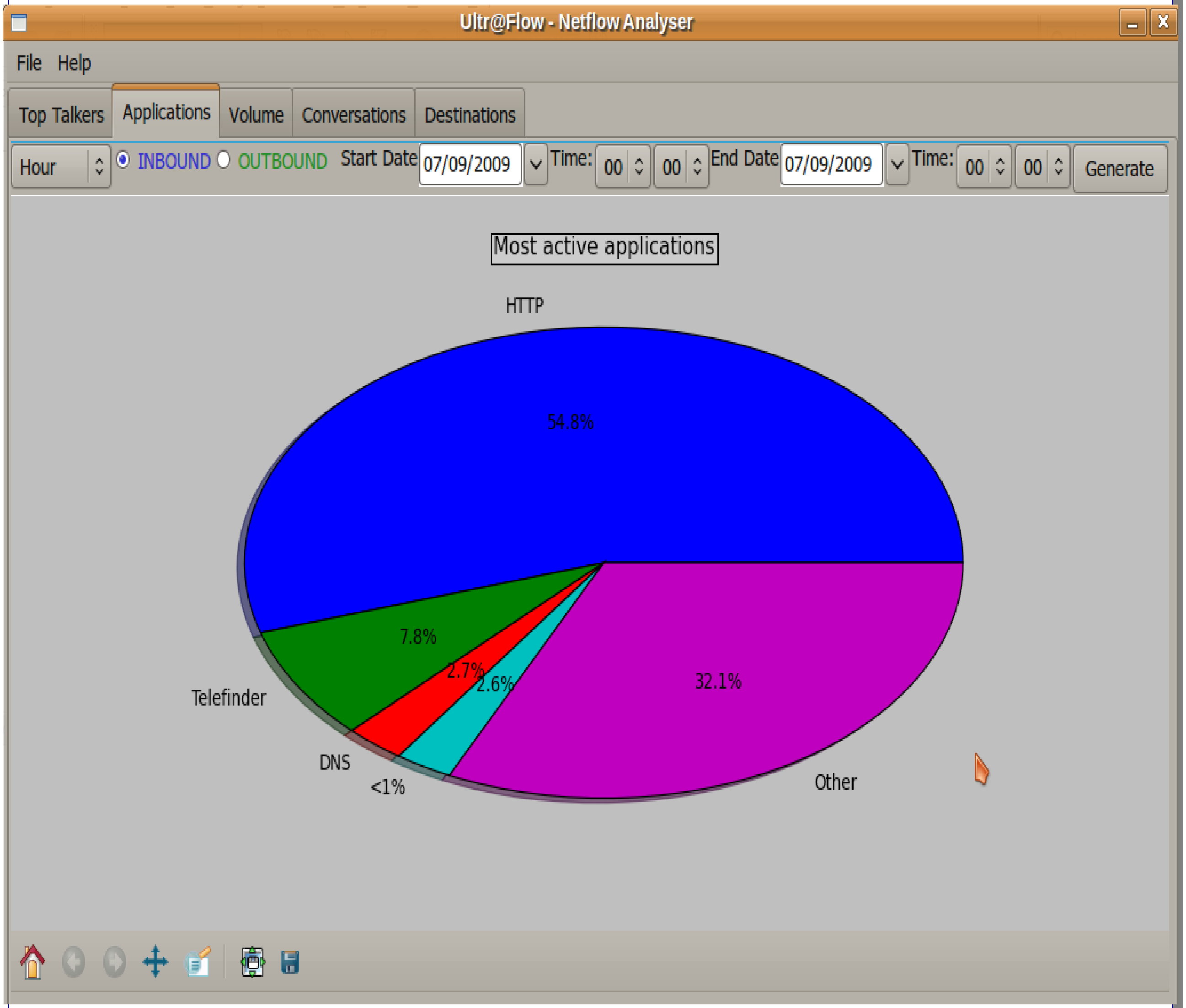


Hour ● INBOUND ○ OUTBOUND Start Date 07/09/2009 Time: 00 00 End Date 07/09/2009 Time: 00 00

Figure 1 - Top Talkers

Top Applications

Bandwidth usage per application gives you visibility into which applications are using the most bandwidth during a given time. Just like the *top talkers* report you can select specific time period like last hour, week, month, year or specific date and time.
From the graph below web traffic accounts for more than half the total traffic.

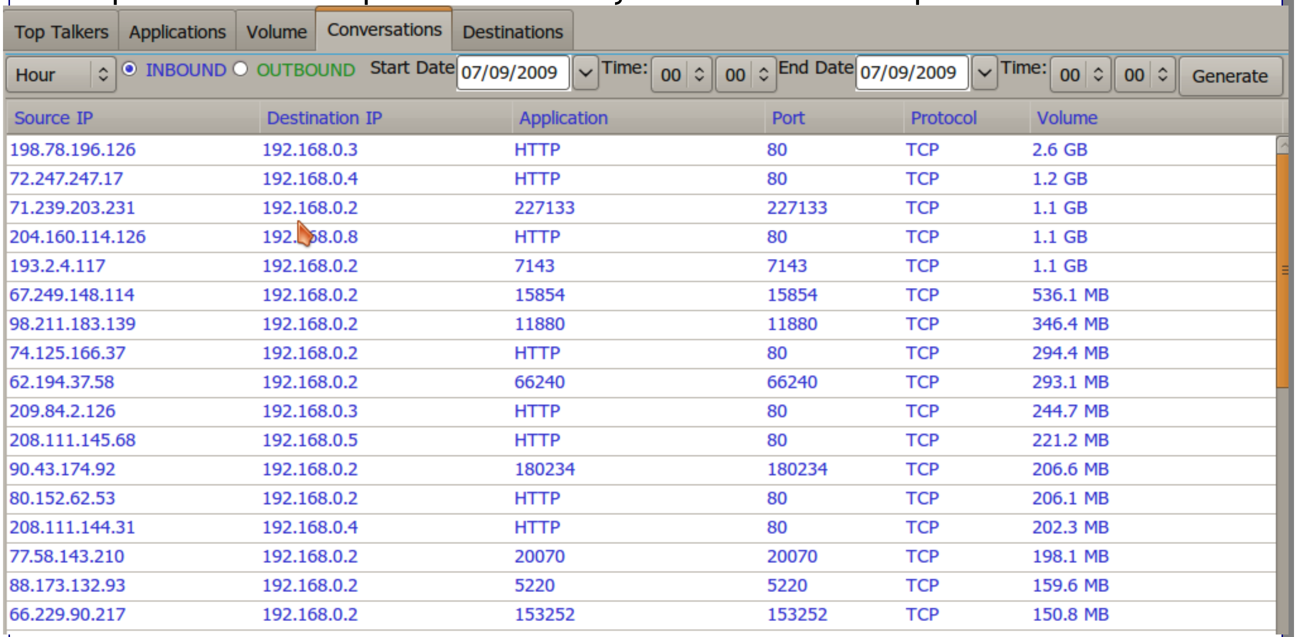


Hour ● INBOUND ○ OUTBOUND Start Date 07/09/2009 Time: 00 00 End Date 07/09/2009 Time: 00 00

Figure 2 – Top Applications

Top Conversations

This report shows the top conversations by traffic volume and protocols involved.

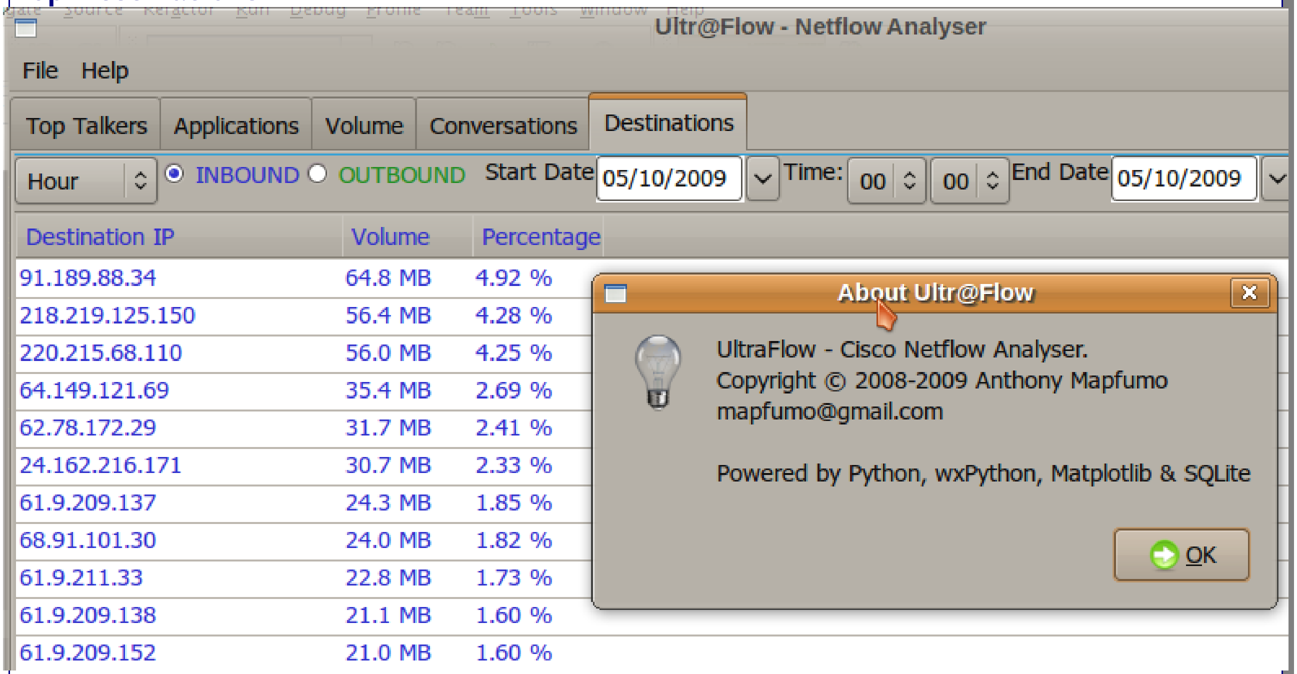


Hour ● INBOUND ○ OUTBOUND Start Date 07/09/2009 Time: 00 00 End Date 07/09/2009 Time: 00 00

Source IP	Destination IP	Application	Port	Protocol	Volume
198.78.196.126	192.168.0.3	HTTP	80	TCP	2.6 GB
72.247.247.17	192.168.0.4	HTTP	80	TCP	1.2 GB
71.239.203.231	192.168.0.2	227133	227133	TCP	1.1 GB
204.160.114.126	192.168.0.8	HTTP	80	TCP	1.1 GB
193.2.4.117	192.168.0.2	7143	7143	TCP	1.1 GB
67.249.148.114	192.168.0.2	15854	15854	TCP	536.1 MB
98.211.183.139	192.168.0.2	11880	11880	TCP	346.4 MB
74.125.166.37	192.168.0.2	HTTP	80	TCP	294.4 MB
62.194.37.58	192.168.0.2	66240	66240	TCP	293.1 MB
209.84.2.126	192.168.0.3	HTTP	80	TCP	244.7 MB
208.111.145.68	192.168.0.5	HTTP	80	TCP	221.2 MB
90.43.174.92	192.168.0.2	180234	180234	TCP	206.6 MB
80.152.62.53	192.168.0.2	HTTP	80	TCP	206.1 MB
208.111.144.31	192.168.0.4	HTTP	80	TCP	202.3 MB
77.58.143.210	192.168.0.2	20070	20070	TCP	198.1 MB
88.173.132.93	192.168.0.2	5220	5220	TCP	159.6 MB
66.229.90.217	192.168.0.2	153252	153252	TCP	150.8 MB

Figure 3 – Top conversations

Top Destinations



Hour ● INBOUND ○ OUTBOUND Start Date 05/10/2009 Time: 00 00 End Date 05/10/2009

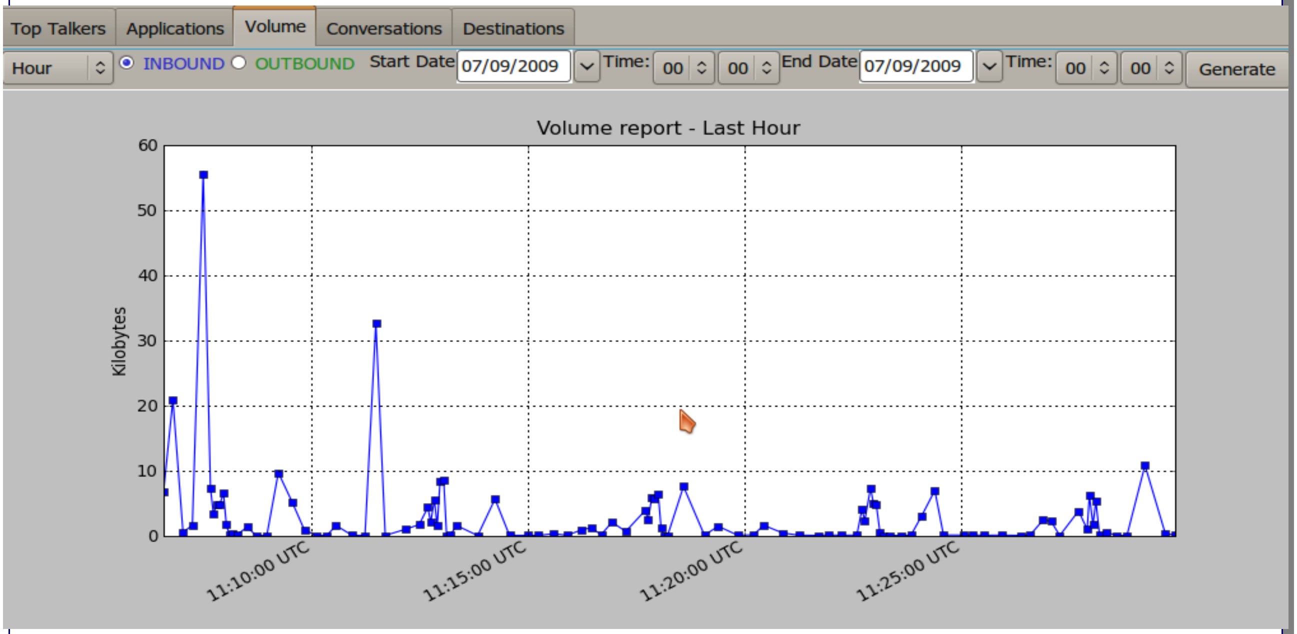
Destination IP	Volume	Percentage
91.189.88.34	64.8 MB	4.92 %
218.219.125.150	56.4 MB	4.28 %
220.215.68.110	56.0 MB	4.25 %
64.149.121.69	35.4 MB	2.69 %
62.78.172.29	31.7 MB	2.41 %
24.162.216.171	30.7 MB	2.33 %
61.9.209.137	24.3 MB	1.85 %
68.91.101.30	24.0 MB	1.82 %
61.9.211.33	22.8 MB	1.73 %
61.9.209.138	21.1 MB	1.60 %
61.9.209.152	21.0 MB	1.60 %

UltraFlow - Cisco Netflow Analyser.
Copyright © 2008-2009 Anthony Mapfumo
mapfumo@gmail.com

Powered by Python, wxPython, Matplotlib & SQLite

Figure 4 - Top Destinations

Traffic Volume Report



Hour ● INBOUND ○ OUTBOUND Start Date 07/09/2009 Time: 00 00 End Date 07/09/2009 Time: 00 00

Figure 5 - Traffic volume for the last hour

Cisco Netflow Architecture

A Netflow system has three major components, a probe/sensor, a collector, and a reporting system.

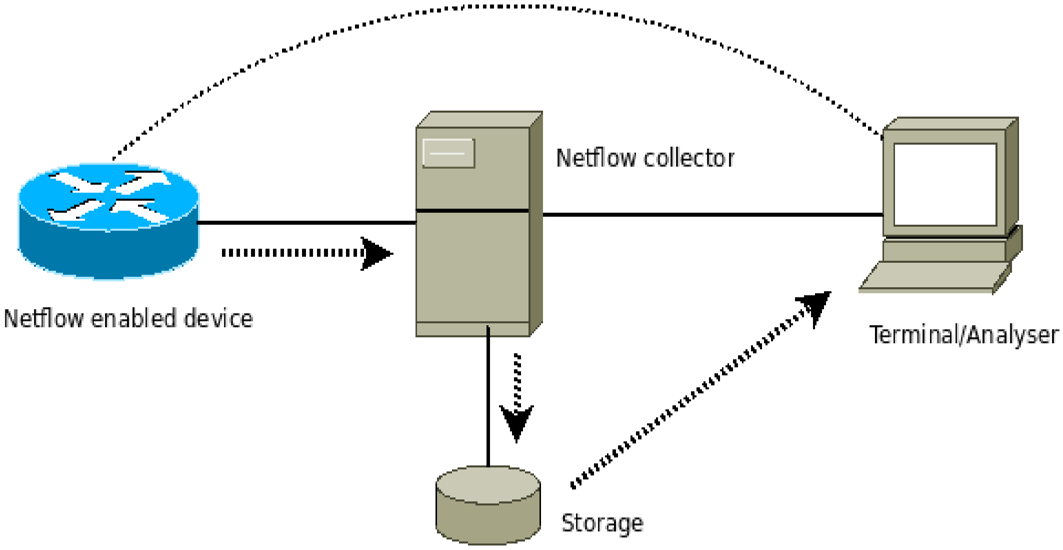


Figure 6 - Cisco Netflow Architecture

A Cisco Netflow can be defined as a 7-tuple key consisting of:-

- Source IP address
- Destination IP address
- Source port for UDP or TCP, 0 for other protocols
- Destination port for UDP or TCP, type and code for ICMP, or 0 for other protocols
- IP protocol
- Ingress interface
- IP Type of Service

NetFlow v5 UDP Header

Name	Description	Offset	Field length (bytes)
Version	Type of record format (5=NetFlow-5)	0	2
Count	Number of flow records contained in the UDP packet (maximum of 30)	2	2
SysUpTime	Value of SysUpTime) when the UDP packet was sent (measured in milliseconds); SysUpTime measures how long the system has been up and running	4	4
Epoch Seconds	Number of seconds between the epoch (January 1, 1970) and the time when the UDP packet was sent. (Epoch is a date used as the "beginning of time" for timestamps. Time values in Unix systems are represented as the number of seconds since the epoch.)	8	4
Nanoseconds	Residual nanoseconds (billionth of a second) after the epoch second	12	4
Flows Seen	Number of flows seen since PacketWise began emitting flow detail records	16	4
Engine Type	User-configurable value (0-255) assigned to the PacketShaper that is emitting flow detail records. Use the setup flowrecords engineType command to assign a type to the unit.	20	1
Engine ID	User-configurable value (0-255) assigned to the PacketShaper that is emitting flow detail records. Use the setup flowrecords engineID command to assign an ID to the unit.	21	1
Sampling Info	n/a	22	2

Flow record format

Bytes	Contents	Description
0-3	srcaddr	Source IP address
4-7	dstaddr	Destination IP address
8-11	nexthop	IP address of next hop router
12-13	input	SNMP index of input interface
14-15	output	SNMP index of output interface
16-19	dPkts	Packets in the flow
20-23	dOctets	Total number of Layer 3 bytes in the packets of the flow
24-27	first	SysUptime at start of flow
28-31	last	SysUptime at the time the last packet of the flow was received
32-33	srcport	TCP/UDP source port number or equivalent
34-35	dstport	TCP/UDP destination port number or equivalent
36	pad1	Unused (zero) bytes
37	tcp_flags	Cumulative OR of TCP flags
38	prot	IP protocol type (for example, TCP = 6; UDP = 17)
39	tos	IP type of service (ToS)
40-41	src_as	Autonomous system number of the source, either origin or peer
42-43	dst_as	Autonomous system number of the destination, either origin or peer
44	src_mask	Source address prefix mask bits
45	dst_mask	Destination address prefix mask bits
46-47	pad2	Unused (zero) bytes

Collector

The Netflow collector is a UDP server written using the Python programming language's Twisted network programming framework. Twisted is flexible, includes a great deal of functionality including mail, web, news, DNS, SSH and database access. It is an event-based and asynchronous framework making it possible to write applications that stay responsive while processing events from multiple network connections without using threads.

The collector listens for Netflow packets on UDP port 9995 (the default Cisco Netflow export port). It can however be changed to another number and the Cisco Netflow devices should also be configured correspondingly as in the following example:-

```
router#configure terminal
router(config)#interface FastEthernet 0/1
router(config-if)#ip route-cache flow
router(config-if)#exit
router(config)#ip flow-export destination 192.168.9.254 9995
router(config)#ip flow-export source FastEthernet 0/1
router(config)#ip flow-export version 5
router(config)#ip flow-cache timeout inactive 15
router(config)#snmp-server ifindex persist
router(config)#^Z
```

Storage

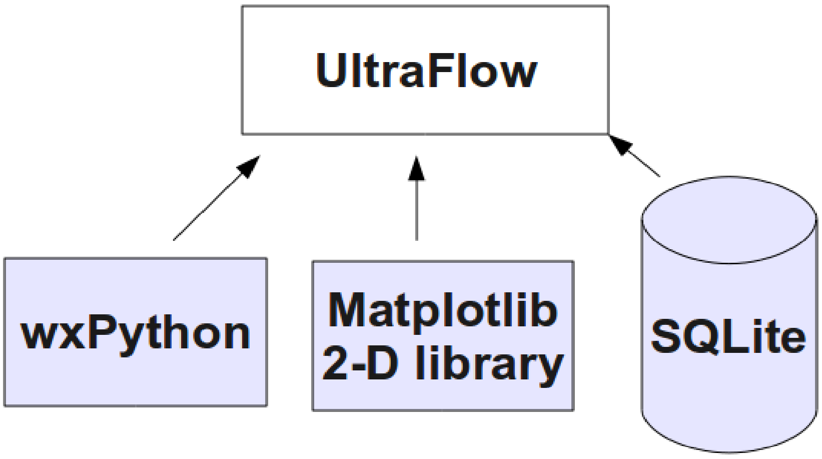
The data storage for this project are pretty modest as such SQLite database is used. SQLite is an open source relational database. It is designed to provide a convenient way for applications to manage data without the overhead sometimes associated with with dedicated relational database management systems like MySQL. It is highly portable, easy to use, compact, efficient and reliable. It implements a large subset of the SQL standard. SQLite eliminates the need for a database daemon process.

```
CREATE TABLE flows (
         router        VARCHAR(64),
         epoch_time    LONG,
         src_addr      VARCHAR(64),
         dst_addr      VARCHAR(64),
         src_port      INTEGER,
         dst_port      INTEGER,
         octets        INTEGER,
         packets       INTEGER,
         protocol      INTEGER,
         tos           INTEGER
);
```

epoch_time: the number of seconds between the epoch (January 1, 1970) and the time when the UDP packet was sent.
Router: the IP address of the Cisco device generating the flows
src_addr: Source IP address
dst_addr: Destination IP address
src_port: TCP/UDP source port number or equivalent
dst_port: TCP/UDP destination port number or equivalent
packets: Packets in the flow
octets: Total number of Layer 3 bytes in the packets of the flow
protocol: IP protocol type (for example, TCP = 6; UDP = 17)
tos: IP type of service (ToS)

Analyser

UltraFlow analyser is based on wxPython which is a graphical user interface toolkit for Python, built upon the wxWidgets C++ toolkit. wxPython is cross platform (runs on Windows, Linux/UNIX, Apple OS X).



Graphs are generated using the Matplotlib library. Matplotlib is a python 2D plotting library which produces publication quality figures using in a variety of hard copy formats and interactive GUI environments across platforms. Matplotlib can be used in python scripts, interactively from the python shell, in web application servers generating dynamic charts, or embedded in GUI applications.

Future Enhancements

MySQL Database

For a highly scalable application a relational database which supports multiple writes and has better security like MySQL would be ideal. MySQL supports a privilege and password system that is very flexible and allows host-based verification. MySQL handles large databases and has maximum table size of 8TB.

C/C++ based collector

For optimum performance the collector will be written in C/C++. C/C++ applications are compiled as compared to interpreted python applications. In most cases this results in better performance for C/C++ applications. Furthermore the open source Gcc compiler can optimise applications for a particular architecture.

Web Based Interface

A web based application Netflow analyser could be accessed from anywhere. It would also be truly cross platform.

References

Netflow - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Netflow

Monitoring network with NetFlow and cflowd
http://brneurosci.org/linuxsetup34.html

wxPython Tutorial
http://wiki.wxpython.org/

Twisted Network Programming Essentials
By Abe Fettig
ISBN: 0-596-10032-9

Core Python Programming, Second Edition
By Wesley J. Chun
ISBN-10: 0-13-226993-7

The Definitive Guide to SQLite
By Michael Owens
ISBN-13: 978-1-59059-673-9

wxPython in Action
NOEL RAPPIN & ROBIN DUNN
ISBN 1-932394-62-1