Extending the IPFIX protocol for better QoS monitoring

Arne Øslebø & Olav Kvittem
UNINETT
7465 Trondheim
Norway
email: arne.oslebo@uninett.no, olav.kvittem@uninett.no

Keywords: netflow, Quality of Service, measurements and monitoring

Background

Network administrators are always striving to get as complete picture as possible of the usage of their network. One technology commonly used for this is flow analysis where information about packets containing the same common attributes are summarised into a flow record. The common attributes are refered to as the flow key and are usually the source and destination IP addresses and ports and the protocol type. The flow records contain information about the flow key as well as information like number of bytes and packets in the flow, start and stop of the flow and source destination AS numbers. The most common flow analysis technology used today is Netflow version 5 from Cisco, and IETF is currently standardizing Netflow in the IPFIX[1] working group.

One problem with these technologies is that while they can provide a lot of information for general use of the network, they provide very little information about the end-to-end quality of service that users get. IPFIX can for example easily give you information about which IP addresses that sends the most traffic, but it can not say anything about the burstiness of this traffic. It can also tell you how much traffic you send to one specific AS number, but it can not tell you who initiated the traffic. Was it someone in your own network that started sending data to this AS, or was it a user in the AS that started to download things from your network.

Implementation

UNINETT wanted to do measurements that makes it possible to answer questions like this as well as other end-to-end QoS questions. Based on passive monitoring cards and technology from the LOBSTER project[1], UNINETT has implemented an IPFIX exporter with several new attributes that provides a lot more details about the quality of service in the flows. This is possible because the IPFIX protocol allows the inclusion of enterprise specific attributes in the flow records.

Some of the new attributes that UNINETT have added are:

pktLenHistogram a histogram of the length of packets in a flow

pktDistHistogram a histogram of the distance between packets in a flow

pktLength[Var/Sum/SumQ] statistical values for length of packets in the flow

pktDist[Var/Sum/SumQ] statistical values for the distance between packets in the flow

direction provides information about who initiated the TCP connection in the flow

reordered number of bytes that are out of sequence in a TCP flow

maxRate[1s/100ms/10ms/1ms] information about the maximum bit rate in the intervals

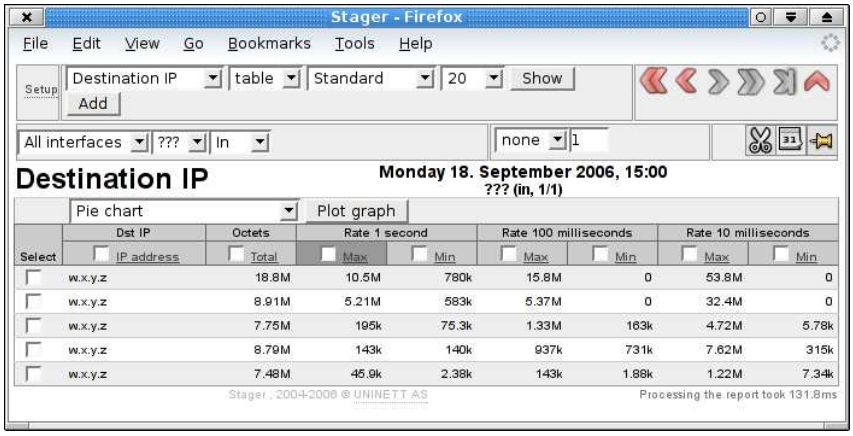

Figure 1: Destination IP address report

minRate[1s/100ms/10ms/1ms] information about the minimum bit rate in the intervals

service looks into the payload of packets to give information about the application that generates the traffic. Recognizes P2P applications like Bittorrent, eDonkey and Gnutella.

rtp[Jitter/LostFraction/LostPackets] looks into RTP packets to provide extra statistics about jitter and lost packets in an RTP flow

These additional attributes makes it possible to create several new interesting Netflow based reports that are not available when using normal Netflow. Figure 1 shows one example. In this figure we can see the top five IP addresses[1] that sends the most traffic on one single observation point. We can see the total amount of traffic that was transferred during the time period displayed as well as maximum and minimum transfer rate in the intervals 1 second, 100 milliseconds and 10 milliseconds. This information makes it possible to say something about the burstiness of the traffic for each IP address. We can see that the top IP address has a maximum transfer rate of 10.5Mbit/s in a 1 second time interval, while the minimum transfer rate is as low as 780Kbit/s. This shows that the traffic from this IP address is quite bursty.

If we look at the fourth IP address, we can see that it has a quite stable transfer rate. The maximum and minimum transfer rate for 1 second only differs by 3Kbit/s.

Performance

The performance of the flow generation itself depends on the parameters collected.The more we dig into the packet, the more cycles are used. The service classification is particularly expensive. Our requirement is to do this in the backbone and the parameters have therefore been chosen for their ease of implementation . The current implementation is able to analyze the traffic on a 2.5Mbps backbone link that is more than 50% loaded. This is done by a PC with 3GHz CPU and a PCI-X bus.

Further work

Further work in this field has the potential of extending the overall knowledge of the quality of the network. The performance numbers can be summarized against any flow parameter, so we will be able to state the throughput specter of a service, an autonomous system - a provider, IP-prefix - a customer. In this way we can potentially better evaluate providers and customer internal and external performance.

[1] The IP addresses are anonymized for privacy reasons

We can also look more detailed into protocols like TCP and RTP to see how they perform in various parts of the network. Service classification is complex and constantly changing due to new protocols and services.To achieve higher speeds like going to 10Gbps we need more CPU and possibly memory bandwidth. Parallelization is a possible way to go.

An early prototype of this work was presented at the NORDUnet2006 conference. At that time only a few simple reports were available and only one single monitoring probe was used for testing. At the time of TNC 2007 it is expected that several monitoring probes that are being deployed as part of the GigaCampus[2] project will be running this IPFIX exporter. There will also be several new reports available.

We do not plan to submit a full paper.

References

[1] IP Flow Information Export (IPFIX) working group, http://www.ietf.org/html.charters/ipfix-charter.html

[1] The LOBSTER project, http://www.ist-lobster.org

[2] The GigaCampus programme, http://www.gigacampus.no/om.en.html

1 Author Biographies

1.1 Arne Øslebø

Arne Øslebø¸ received a M.Sc. from the Norwegian University of Science and Technology (NTNU) in 1997. From 1997 to 2001 he worked as a research fellow at NTNU working with network management. He started working for UNINETT in 2001 where he works on network management and monitoring. For the last few years he has been heavily involved in IST projects like SCAMPI and LOBSTER.

1.2 Olav Kvittem

Olav Kvittem, received a M.Sc. in Computer Science from the Norwegian University of Science and Technology (NTNU) in 1997. He has worked 5 years on systems programming for mainframes, 9 years in research on networking, languages and parallel operating systems at SINTEF and 18 years in academic networking on network level issues for UNINETT. Presently leading the Research and Development Division entitled as Chief Technical Officer. Has been active in Terena networking groups and also on the Terena Technical Committee for 7 years.