Traffic Characterization (Assignment 2)

Fabrice SERVAIS, Laurent VANOSMAEL (Group 6) - Master student in Computer Science and Engineering, ULG

***Abstract*—This report aims at describing the traffic at ULg during a 6-hour period by analyzing Netflow records and extracting packet sizes, flow information, port utilisation and prefixes involved in the communications captured.**

I. INTRODUCTION

The following article will try to characterize the traffic going through an endpoint of the University of Liege's network connected to the Belnet network. The goal will be to analyze the behaviour of the traffic following the guidelines given in the assignment while using a huge dataset to study efficiently.

The Netflow records have been collected using `nfdump` during 6 hours and then anonymized. This dataset was stored in a CSV file having a size of nearly 30GB. This report will thus first overview the tools and techniques used to analyze efficiently such a dataset. It will then follow the guidelines of the assignment by explaining the average packet size, analyzing the flow durations and size, studying ports used in the communications, as well as the prefixes participating.

II. DATASET MANAGEMENT

According to the size of the dataset, it is obvious that it can't fit all at once in the memory. However, the method `read_csv` from the `pandas` library provides a mechanism to read such big files by **chunks** of a given number of lines. The script thus iterates one chunk at a time and saves the needed value for later computations, i.e:

- Number of bytes of each flow
- Number of packets of each flow
- Duration of each flow
- Source address
- Destination address
- Source port
- Destination port

Concerning the source and destination addresses and ports, they are aggregated at each iteration to consider the total number of bytes and packets by address and port.

Also, another principle used to increase the efficiency is that the usage of the `pandas` library through its data structures and algorithms was always privileged against others due to the high-performance capabilities of `pandas` for handling big data.

III. PACKET SIZE

The number of bytes and packets for each flow were saved when reading the dataset. The average packet size for each flow is then computed. The average packet size is computed by taking the mean of the average packet sizes for each flow. From those computations, the mean packet size of the dataset is 219.55 bytes.

IV. FLOWS ANALYSIS

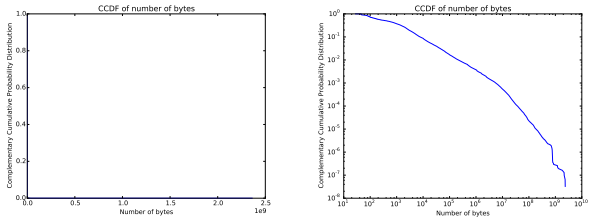

Fig. 1. CCDF of the bytes traffic per flow (left: linear, right: log scale)

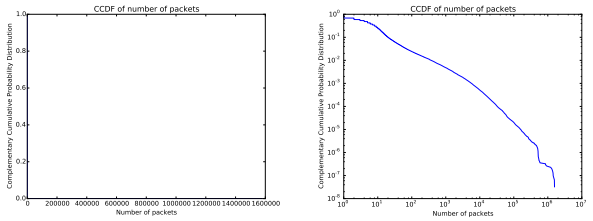

Fig. 2. CCDF of the packet traffic per flow (left: linear, right: log scale)

The graphs of the complementary cumulative distribution frequency of the traffic size per flow (Fig.1) and the number of packet per flow (Fig.2) have quite the same shape.

50 % of the flows have a size of less than 417 bytes while 99% of the flows have a size of less than 214KB. On the other hand, 50% of the flows contains less than 4 packets and 99% less than 371 packets. It shows that most of the flows are light and small in terms of packets. This is graphically shown, where we can see that the probability that a flow has a size above 10 KB is less than 0.1%. In a similar way, the probability that a flow has a size above 50 packets is less than 0.1%.

Knowing that most of the traffic are Web based traffic (HTTP or HTTPS), see Section IV for more information, it could be deduced than maybe most of the web flows are not *Keep Alive* flows.

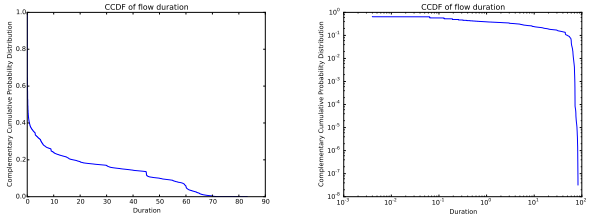

Fig. 3. CCDF of the packet traffic per flow (left: linear, right: log scale)

The graph Fig.3 shows also that most of the flows are short. Indeed 50% of the flows last less than 0.15 second and 90%

less than 50 seconds. This is more visible in the log scale graph where one can more clearly see the durations less than 1 second.

As shown in Fig. 1 and 2, the graphs with a linear scale don't show anything. Indeed, the values cover a wide range of orders of magnitude. Log scale graphs are more suitable to represent such kind of values.

V. Port number ranking

The following sections will present the results of an analysis based on the sending and the receiving port numbers. In the following results, the fraction of the traffic is computed as the fraction of the traffic size in bytes.

It is also important to remember that port > 1023 are not classified as well-known port and are often used for other applications.

A. Sender Ports

The top 10 sending port are given in Table I and represented in a pie chart in Fig. 4.

TABLE I
Top 10 senders port

Port	Main Usage	Fraction of traffic
443	HTTPS	33.86%
80	HTTP	30.31%
8080	HTTP Alternate	6.00%
22	SSH	3.89%
8443	PCsync HTTPS	3.25%
4500	IP Sec Microsoft	1.68%
61817	XSan file-system access	1.55%
993	Imap4 over TLS/SSL	1.07%
40018	Windows Update and Auth check	0.92%
63099	Xsan file-system	0.8%
	Others	16.57 %

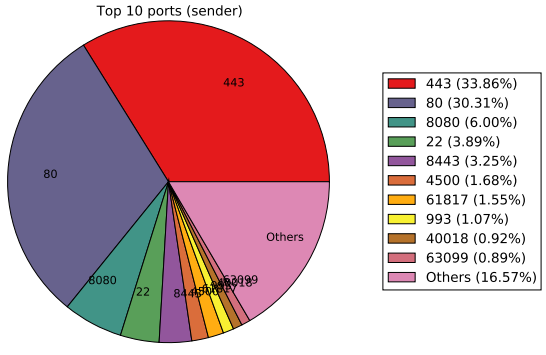

Fig. 4. Top 10 port sender

The table shows that the biggest part of the traffic are sent from the top 10 ports. Among them, we can see that web traffic applications use the largest amount of traffic, in the top 5 port 4 are related to web browsering.

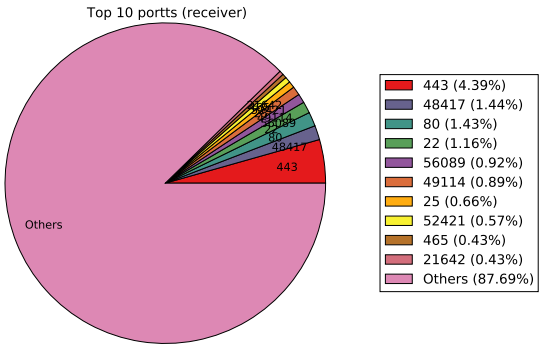

Fig. 5. Top 10 receiver port

TABLE II
Top 10 destination port

Port	Main Usage	Fraction of traffic
443	HTTPS	4.39%
48417	Unknown	1.44%
80	HTTP	1.43%
22	SSH	1.16%
56089	XSan file-system	0.92%
49114	Unknown	0.89%
25	SMTP	0.66%
52421	X11 Toolkit	0.57%
465	SMTP with SSL	0.43%
21642	Unknown	0.43%
	Others	87.69%

B. Destination port

The top 10 receiving port are given in Table II and represented in a pie chart in Fig. 5.

From the pie chart of the receiver ports, the "other ports" category is largely predominant. This is due to the fact that the percentage of traffic at the receiver side is more distributed amongst the ports, even if the web ports (443 and 80) are at the top of the ranking, there are a lot of unknown ports in the ranking.

Indeed, since most of the traffic (in bytes) is due to a reply from a server, the client ports appear for a bigger part of the traffic. Also, the port on the client side is randomly chosen when sending the request, that explains the wide range of ports and the distribution of the traffic amongst more port numbers.

VI. Prefix aggregation and analysis

A. Method

Since the prefixes were not available in the dataset, the goal is to assign guess the prefixes involved in the flow.

As first try of algorithm, the first step was to order the dataset by IP address, so that the addresses were intrinsically grouped. The next was thus to find those groups. To do so, a distance measure between addresses was need. One natural solution is to define the distance between 2 IP addresses as the number of IP addresses between them. Using that, we can iterate through the ordered IP addresses. We thus define the separation on 2 groups when the distance between lest last IP

of the first group and the fist IP of the second IP is greater than a threshold. When the groups have been defined, the last step is to find the prefix. To do so, one can just find the longest common (binary) prefix and aggregate the number of bytes by the prefix found.

This methodology however is really too slow and is not applicable to the 30GB dataset. As a replacement solution, the prefix /24 was attributed to the IPv4 addresses and the /32 to the IPv6 addresses as an estimation as they are respectively the most common prefixes. [1]

B. Results

The result coming from the aggregation of the traffic into prefixes is shown for different percentages of the most popular prefixes in Table III. These results show that most of the traffic (almost the entire traffc) comes from the same top 10% /24 subnets.

% of top prefixes	**% of the traffic**
0.1%	87.29071%
1%	97.64789%
10%	99.91141%

TABLE III
PERCENTAGE OF THE TRAFFIC FOR TOP PREFIXES

C. The 92.106.195.0/24 case

The results of the analysis of the traffic of the 92.106.195.0/24 block are shown in the TABLE IV.

	Percentage of packets	**Percentage of bytes**
Sent by 92.106.195.0/24	0.053%	0.065%
Sent to 92.106.195.0/24	0.038%	0.025%

TABLE IV
TRAFFIC OF THE 92.106.195.0/24 ADDRESS BLOCK

REFERENCES

[1] Devin Bayer, "*Visibility of Prefix Lengths in IPv4 and IPv6*", Oct 2009. https://labs.ripe.net/Members/dbayer/visibility-of-prefix-lengths