INFO8011: Network Infrastructures

Team 01: Jérôme BAYAUX, Joachim PAQUAY

I. INTRODUCTION

In this project, one analyzed six hours of NetFlow records captured from a router in the Université de Liège directly connected on the Belnet network.

II. QUESTION 1

As one can see on the figure 1, there is a huge proportion of small and big packets. Indeed one can see there is almost 40% of packets with at most 200 bytes and 40% of packets with 1300 bytes and more.

The smallest packets have a size of 40 bytes because even if one sends no data with its packet, the TCP header is 20 bytes long and the IP header is 20 bytes long too. Thus, one has a total of 40 bytes minimum.

One can deduce that the small packets comes probably from, for example, a TCP connection (which has a minimum size of 40 bytes) and the big packets from a download, it is limited to 1500 because one cannot overpass the IEEE 802.3 Ethernet norm[1]. This shows the fact that medium packets is not much that used because when things are download for instance, there is no point to send medium packets when you can send large one.

The average packet size across all the traffic in the trace is 840.05 bytes and it corresponds more or less to 0.5 on the y-axis on the figure 1. There is no surprise there. However, this average packet size is not really representative of the traffic. Actually, packets are either quite large or small but they rarely get a size close from 840 bytes.

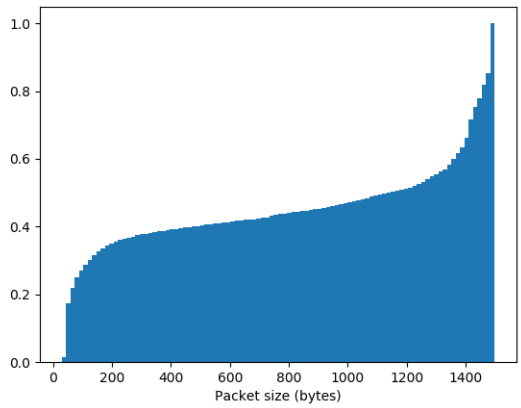

Fig. 1: Packet size distribution as a Cumulative Distribution Function across all traffic in the trace

[1] https://fr.wikipedia.org/wiki/Ethernet

III. QUESTION 2

On figure 2, one can see that majority of flows last for less than 20 seconds. Moreover, there are more than 60 percent of flows with a duration of a few seconds. This can be explained by the fact that nowadays, if there are not any problems in the path from source to destination (for instance congestion, broken links, etc.) packets travel very fast. According to this hypothesis, the minority of flows that have longer durations are probably due to network problems or some network interference like congestion, broken links or limited bandwidth.

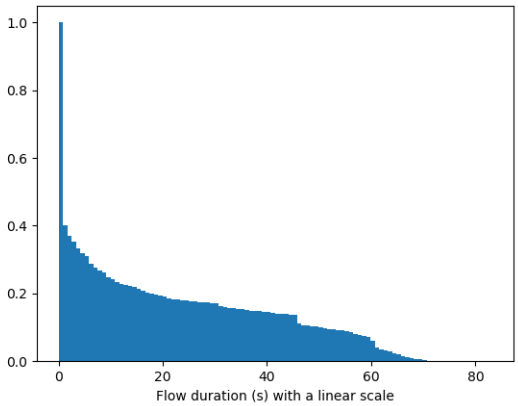

Fig. 2: CCDF of flow durations across all traffic in the trace with a linear scale

Although figures 3 and 4 give a better view of respectively flow sizes in packets and flow sizes in bytes, the graphics are still not ideal because of the logarithmic scale on the ordinate axis.

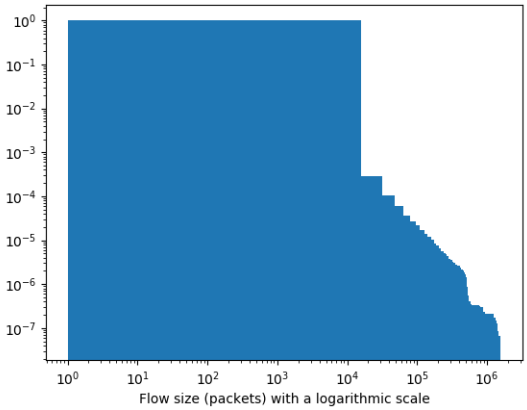

Fig. 3: CCDF of flow sizes (in packets) across all traffic in the trace with a logarithmic scale

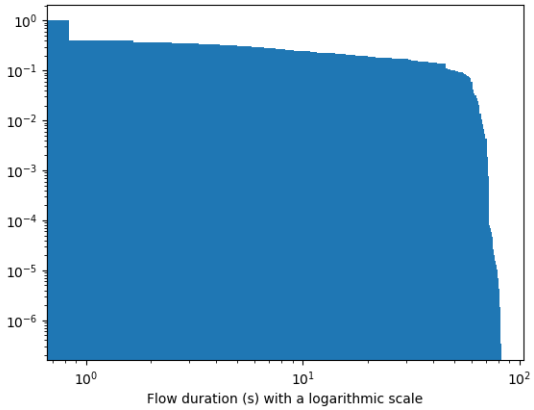

Fig. 5: CCDF of flow durations across all traffic in the trace with a logarithmic scale

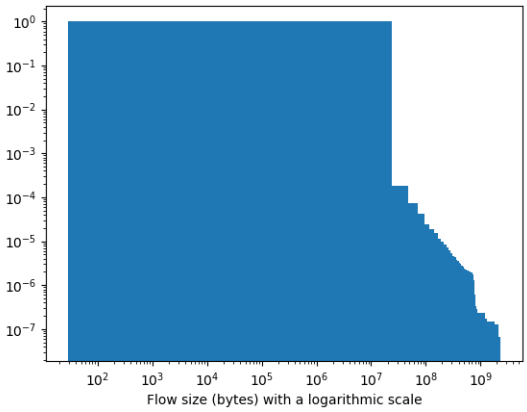

Fig. 4: CCDF of flow sizes (in bytes) across all traffic in the trace with a logarithmic scale

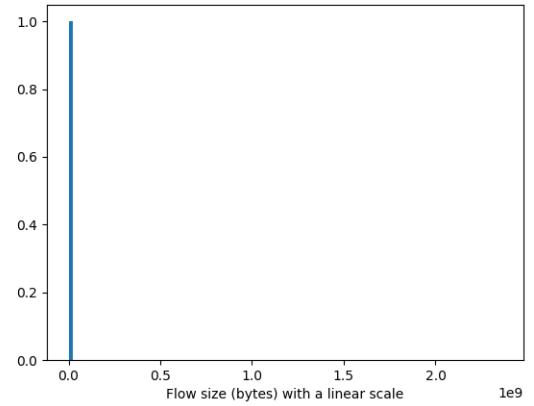

Fig. 6: CCDF of flow sizes (in bytes) across all traffic in the trace with a linear scale

Figures 6 and 7 are not really useful here. Indeed, with a linear scale one can just see a simple vertical line which is not representative of anything. This is why it is better to use logarithmic scales in those cases. In fact, logarithmic scales can show widely spread values in a short interval, they are well suited to highlight the different orders of magnitude. Despite this, it can sometimes be more appropriate to use a linear scale. For example, figure 2 is much more interesting than figure 5.

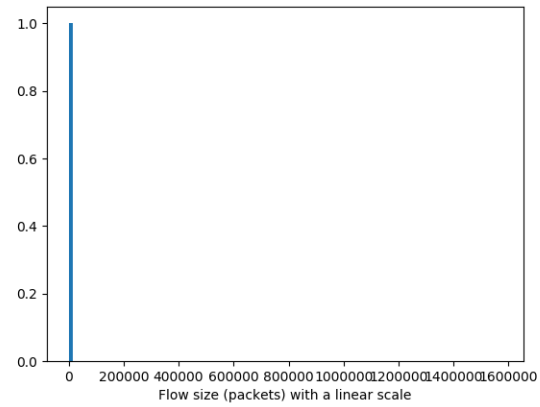

Fig. 7: CCDF of flow sizes (in packets) across all traffic in the trace with a linear scale

IV. Question 3

As one can see in the table I there is two very significant source port numbers which are 443 (HTTPS) and 80 (HTTP) which mainly correspond to web services and web communication such as internet pages,...
An other import port is the port 8080 which corresponds to an alternative of the port 80 so it means web communication too. So for the biggest majority, this corresponds to web communication.

One can also look at the port 22 which corresponds to Secure Shell (SSH), secure logins, file transfers (scp, sftp) and port forwarding.

sp	ibyt	percentage
443	1082609028484	33.87
80	969207034633	30.32
8080	191840531703	6
22	124459603816	3.89
8443	103859067410	3.25
4500	53631334018	1.68
61817	49703575265	1.55
993	34229809815	1.07
40018	29349232006	0.92
63099	28511972053	0.89

TABLE I: Top-ten port numbers by sender traffic volume

An important port is the port 48417[2] which corresponds mainly to TCP/UDP informations and the application will probably be a TCP connection.

dp	ibyt	percentage
443	140345036504	4.39
48417	45957956710	1.44
80	45609890407	1.43
22	37092366770	1.16
56089	29255701208	0.92
49114	28542196806	089
25	21030082789	0.66
52421	18218580937	0.57
465	13715114883	0.43
21642	13693430334	0.43

TABLE II: Top-ten port numbers by receiver traffic volume

Http and https are obviously the two protocols responsible for the majority of the traffic volume. One expects then to see ports 80 and 443 in the top-ten port numbers by receiver traffic volume as servers listen on those particular ports. However, if one looks more closely to tables I and II, one observes that the top-ten port numbers by receiver traffic volume is composed of more random ports than top-ten port numbers by sender traffic volume. Furthermore, in table II, ports 80 and 443 account only for less than 6% of the traffic volume, compared to more than 60% in table I. This is certainly due to the fact that the amount of bytes sent as an answer to an http (or https) request is much bigger than the size of the request itself. Thus, ports 80 and 443 that appear in table I are servers ports by which they forward responses to http or https resquests.

[2] https://fr.wikipedia.org/wiki/Liste_de_ports_logiciels

V. Question 4

For this question and the following one, the IPV6 addresses found in the file has been discarded for the sake of ease. This does not affect the results that are presented in this section because IPV6 addresses only appear in less than 3.5 millions of lines among more or less 93 millions in total.

In order to plot the figure 8 only the 100 biggest IP prefixes by traffic volume were took because as one can see on the figure, there is no point to take more than the 100 biggest because one can clearly see that the % of traffic volume is very small and is very close to 0.
Table III shows the 10 biggest IP prefixes by traffic volume represented in the figure 8.

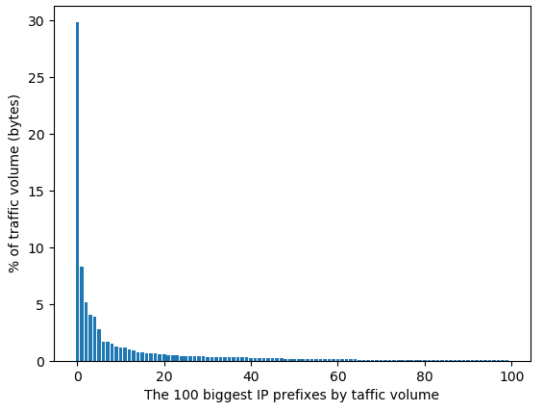

Fig. 8: Traffic volumes based on the source IP prefix

	Sender prefixes	Number of bytes	% of traffic (bytes)
1	139.91.0.0	954201528253	29.85
2	178.2.0.0	266294353932	8.33
3	117.249.0.0	166329545688	5.2
4	167.76.0.0	131581562634	4.12
5	115.156.0.0	124101050356	3.88
6	198.69.0.0	90536554870	2.83
7	218.66.0.0	55581755280	1.74
8	31.2.0.0	54674233476	1.71
9	157.225.0.0	49473082054	1.55
10	107.144.0.0	42133894096	1.32

TABLE III: Percentage of the 10 biggest IP prefixes by traffic volume in bytes and the number of bytes associated

The percentages of traffic volume in bytes that comes with the most popular 0.1%, 1% and 10% of source IP prefixes are represented in the table IV.

	% of the traffic volume in bytes	
	All the traffic	Prefixes that have a positive mask length
Most popular 0.1%	74.35	31.23
Most popular 1%	92.91	64.62
Most popular 10%	96.29	91.36

TABLE IV: Percentage of the traffic volume in bytes that comes with the most popular 0.1%, 1% and 10% of source IP prefixes

In table IV The difference between the column with all the traffic and the column with a positive mask length is simply given by the fact that the column with all the traffic contains the masks with a length 0 when the other does not contain the masks with a length 0.
According to us, not having a positive mask length means not having a significant prefix. Then, in order to determine where the mask has a length 0 or not, one supposition is simply to filter by the percentage of the total traffic (in bytes). To get those results, the source IP addresses with a percentage > 0.001 were considered as having a mask length > 0 when the others were considered has having a mask length of 0.

VI. Question 5

In order to find the prefix of the university of Liège, one can suppose it corresponds to the biggest sender and receiver prefix which is 139.91.0.0/16 according to the table III.

In order to determine the Montefiore and RUN prefixes, one can take the prefix 139.91.0.0/16 of the university and check were the prefixes 139.91.X.0/24 might correspond to a plausible response.

When analysing all the percentages in the .csv file generated by the Q5's functions in the python file, one can guess that Montefiore might have the anonymous prefix 139.91.64.0 when looking a the results. Indeed, Montefiore is probably not the biggest subnet of the university but one of the biggest and it will probably have a biggest receiver percentage than sender percentage. By following this assumption, one can deduce one possible prefixe for Montefiore which is given in the table V. But other prefixes might work too, for example 139.91.108.0, 139.91.9.0,...

For RUN, an assumption could be that this section will send a lot of packets but with very few bytes and will also "download" or receive a lot of response because they are doing research in network so they might scan the network, doing a lot of pings,... So a plausible prefix might be 139.91.154.0 (given in the table V) but when looking at the .csv file generated by the Q5's functions, one could find other possibilities such as 139.91.7.0, 139.91.31.0,...

		bytes		Packets	
	Prefix	Sender	Receiver	Sender	Receiver
Montefiore	139.91.64.0	0.61	1.1	0.67	0.79
RUN	139.91.154.0	0.29	4.59	1.52	3.18

TABLE V: Percentage of the total traffic by bytes and packets in the trace sent by Montefiore and RUN

VII. Conclusion

To conclude, it was really interesting to analyse real traffic and see what is happening in the network. The analysis have allowed us to verify the theory we learned in our previous networking courses but they have also made us realise that nowadays, the amount of data we have to process is really big.