# INFO8011: Network Performance and Measurements

Team 14: Tom CRASSET, Thibault RENAUD

## I. INTRODUCTION

In this report, six hours of NetFlow records ($\approx$ 30 GB of data) captured from a router in the Université de Liège directly connected on the Belnet network will be analyzed. Our work aims to link some theoretical concepts to the empirical results obtained with the Netflow records.

## II. PACKET SIZE DISTRIBUTION

The Figure 1 represents the cumulative distribution function (CDF) of the packet size across all traffic in the trace. Note that the size of a packet corresponds to average number of bytes overall packets belonging to a same flow.

To determine the number of packets having the same size, we have multiplied the number of packets belonging to a flow by the average packet size (in bytes) of this flow. Then we have gathered the same sizes together and we have summed up the number of packet of each same size.

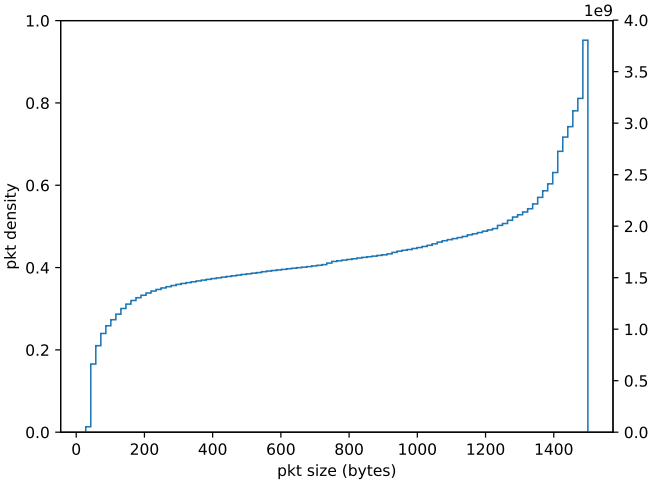



Fig. 1: CDF of packet size

The minimum size of a packet is 28 bytes, which is not the minimum value one could expect. Indeed, 40 bytes is the minimum packet size for TCP. One could conclude that a problem happened during the dump operation.

The maximum packet size is 1500 bytes, which is the maximum Ethernet payload size from TCP implementations that use path MTU discovery. The average size of a packet is 724 bytes. We could have expected a value close to half of 1500 bytes because there is almost the same amount of small packets ([0; 200]) than of large ones ([1200; 1400]), as can be seen on the Figure 1. There are just a few packets for which the size is between 200 bytes and 1200 bytes.

## III. PACKET FLOWS

In this section, one will find the plots of the complementary cumulative distribution function (CCDF) of flow duration (in second) and of flow sizes (in number of bytes, and in number of packets). It is sometimes useful to have plots with logarithmic scale when there are a lot of small packets and a few very large ones, as you can see on Figure 3, 4 (b), and 5 (b).

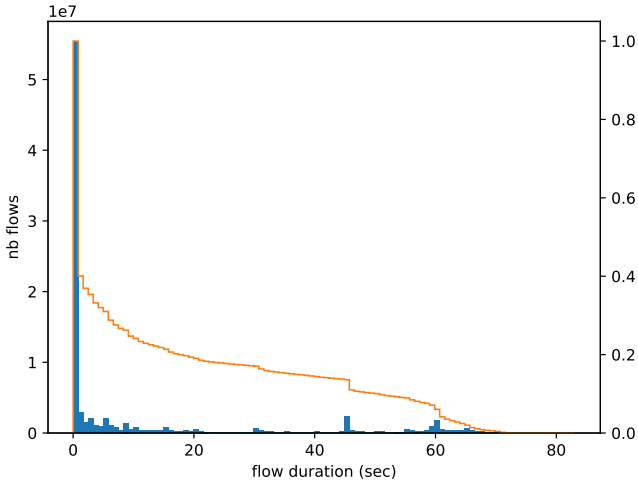



Fig. 2: CCDF of flow duration (in sec)

We notice in the Figure 2 that the majority of the flows duration are less than 1 second, and only 40% of the total number of flows lasts more than 1 second. In one second, the number of bytes that can be transmitted is of the order of MBs, which is quite large. One can easily imagine that flows containing a lot of packets of size 1500 (bytes) are part of the 40%. The majority of the flows contains only a few MBs that can be transmitted in maximum 1 second. It would be interesting to have a tool in NetFlow to get more precise records regarding the flow duration (for instance in ms).

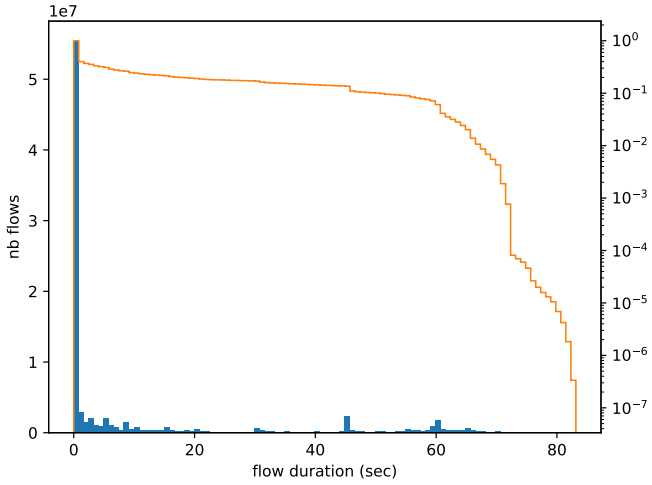
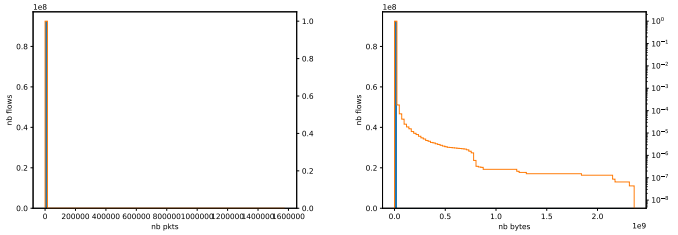


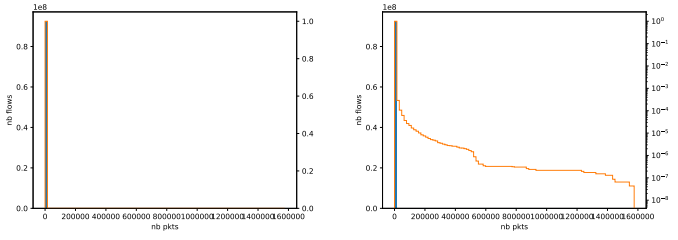

Fig. 3: CCDF of flow duration (in sec)

As can be seen in Figure 3, among the 40% of the flows for which the flow duration is above 1 second, flows with duration above 60 seconds are very rare and drop very quickly to 0.



(a) linear axis (b) log axis

Fig. 4: CCDF of nb of bytes in a flow



(a) linear axis (b) log axis

Fig. 5: CCDF of nb of packets in a flow

One can notice that Figure 4 (b) and 5 (b) are very similar. If the number of packets in a flow increases, so does the number of bytes. This fact provides us 2 pieces of information: obviously a packet contains some bytes, but more importantly if we take a point on both graph (density $\approx 10^{-6}$) then we see in equation (1)

$$\frac{nb\_bytes}{nb\_packets} \approx \frac{10^9}{700000} = 1429 \approx 1500. \quad (1)$$

So the packets in the flows are mainly of size of the MTU packets. This can be verified by taking another point on both graphs. The higher the number of packets in a flow, the higher the number of bytes in this flow, and this number is approximately equal to $1500 \times nb\_packets$.

## IV. Traffic Volume Analysis w.r.t. port numbers

You can observe the top-ten sender and receiver port numbers in Table I and II, as well as the corresponding service name. You can also find two pie charts, Figure 7 and 8, to have a good visualization of the distribution of the amount of traffic among the different port numbers.

Without any surprise, the port number that sends the highest amount of data is the 443. It is the https service. It is also the one that receives the highest amount of data. It is followed by the http service, which sends half the amount of data https sends.

| Sender Port # | Service name | Traffic Volume (in GB) | Contribution (%) |
|---|---|---|---|
| 443 | https | 1082.609028484 | 33.87 |
| 80 | http | 969.207034633 | 30.32 |
| 8080 | http-alt | 191.840531703 | 6 |
| 22 | ssh | 124.459603816 | 3.9 |
| 8443 | pcsync-https | 103.859067410 | 3.2 |
| 4500 | ipsec-nat-t | 53.631334018 | 1.68 |
| 61817 | / | 49.703575265 | 1.55 |
| 993 | imaps | 34.229809815 | 1.07 |
| 40018 | / | 29.349232006 | 0.92 |
| 63099 | / | 28.511972053 | 0.089 |
| other | / | 529.219559934 | 16.56 |

TABLE I: Top-ten sender port numbers

| Receiver Port # | Service name | Traffic Volume (in GB) | Contribution (%) |
|---|---|---|---|
| 443 | https | 140.345036504 | 0.043904 |
| 48417 | / | 45.957956710 | 0.014377 |
| 80 | http | 45.609890407 | 0.014268 |
| 22 | ssh | 37.092366770 | 0.011604 |
| 56089 | / | 29.255701208 | 0.009152 |
| 49114 | / | 28.542196806 | 0.008929 |
| 25 | smtp | 21.030082789 | 0.006579 |
| 52421 | / | 18.218580937 | 0.005699 |
| 465 | urd | 13.715114883 | 0.004291 |
| 21642 | / | 13.693430334 | 0.004284 |
| others | / | 2803.160391789 | 0.876914 |

TABLE II: Top-ten receiver port numbers

We can also notice that the ssh port is also often used, either for sending or receiving data. This is not surprising since ssh is a popular tool to access another computer through the network.

There are nevertheless some port numbers that receive a lot of data but that are not in the top-ten of the senders. For example, one can easily imagine a video port that must receive a lot of information but that does not have to send a lot of data back.

## V. Traffic Volume Analysis w.r.t. IP addresses

The task at hand is to aggregate and plot the traffic volumes based on the source IP prefix.

First, to aggregate, we used a technique seen in [1]. This paper uses a tree based approach to aggregate the IP addresses into subnets and then further into bigger networks. Iterating over the dataset, we construct the N-ary tree using the IP addresses as leaves. From these leaves, we compute the subnetwork of a given prefix, which we add to the tree. The prefixes that were chosen were multiples of 8 going from a a value of 24 to 8, thus having, from the root of the tree to the bottom the following prefixes: [8, 16, 24]. These values were chosen because of the particularity that the prefix of the Uliege subnet is 16 and those of RUN and Montefiore are 24. A very simple example from the beginning of the trace can be seen on Figure (6) using prefixes of 8 and 6.

Fig. 6: Tree aggregating the IP addresses by prefixes

From this tree, we could compute the top 10%, the top 1% and the top 0.1% of subnetworks by traffic volume. Because of the large amount of data, the computation was restricted to a limited portion of the dataset (7%) and thus the following values in in Table (III) are only estimations of the real traffic passing through these networks.

As you can see, we obviously have less Top 10 networks than networks from the other Top categories. The list was limited to 29 entries, which is the number of Top 1% networks, there were more TOP 0.1% networks than shown. Notice also that network 139.0.0.0/8 and 139.91.0.0/16 are in all three columns, which showcases the method we chose works well.

## VI. Montefiore and RUN traffic

In order to determine the IP prefix of the university of Liège, we have use the fact that the records have been taken from a router on the edge of the University network. Therefore, most of the traffic going in and out must belong to the University of Liège. Thus, we have chosen the subnet (with the prefix of 16) that gathers the highest amount of traffic.

We know that the Uliège owns X.X.Y.B/16 address block. We would like to know the fraction of the traffic (by bytes and by packets) in the trace sent/received by/at Montefiore and RUN. However, we do not know which subnet inside the

| | Top 10% | Top 1% | Top 0.1 % |
|---|---|---|---|
| 1 | 139.0.0.0/8 | 139.0.0.0/8 | 77.0.0.0/8 |
| 2 | 139.91.0.0/16 | 139.91.0.0/16 | 139.0.0.0/8 |
| 3 | 212.0.0.0/8 | 110.0.0.0/8 | 139.91.0.0/16 |
| 4 | 212.74.0.0/16 | 198.0.0.0/8 | 199.0.0.0/8 |
| 5 | | 198.69.0.0/16 | 199.80.0.0/16 |
| 6 | | 103.0.0.0/8 | 199.158.0.0/16 |
| 7 | | 178.0.0.0/8 | 199.242.0.0/16 |
| 8 | | 178.2.0.0/16 | 110.0.0.0/8 |
| 9 | | 117.0.0.0/8 | 110.206.0.0/16 |
| 10 | | 117.249.0.0/16 | 110.200.0.0/16 |
| 11 | | 24.0.0.0/8 | 110.4.0.0/16 |
| 12 | | 24.194.0.0/16 | 96.0.0.0/8 |
| 13 | | 115.0.0.0/8 | 96.144.0.0/16 |
| 14 | | 115.156.0.0/16 | 104.0.0.0/8 |
| 15 | | 31.0.0.0/8 | 104.99.0.0/16 |
| 16 | | 31.2.0.0/16 | 198.0.0.0/8 |
| 17 | | 157.0.0.0/8 | 198.69.0.0/16 |
| 18 | | 157.225.0.0/16 | 10.0.0.0/8 |
| 19 | | 218.0.0.0/8 | 10.119.0.0/16 |
| 20 | | 218.66.0.0/16 | 10.109.0.0/16 |
| 21 | | 212.0.0.0/8 | 124.0.0.0/8 |
| 22 | | 212.74.0.0/16 | 124.209.0.0/16 |
| 23 | | 167.0.0.0/8 | 106.0.0.0/8 |
| 24 | | 167.76.0.0/16 | 106.3.0.0/16 |
| 25 | | 141.0.0.0/8 | 103.0.0.0/8 |
| 26 | | 141.7.0.0/16 | 103.182.0.0/16 |
| 27 | | 92.0.0.0/8 | 103.203.0.0/16 |
| 28 | | 92.106.0.0/16 | 178.0.0.0/8 |
| 29 | | 42.0.0.0/8 | 178.2.0.0/16 |

TABLE III: Top subnetworks by traffic volume

subnet of Uliège corresponds to Montefiore's one and which one corresponds to RUN's one.

From Table (III), we guess that the Uliege network was anonymized using the address 139.91.0.0/16, because the fact that this address is in the Top 10 % of traffic and has a prefix of 16, just like the real Uliege network. Using this knowledge, we would have decided to get the subnets of the form the prefix 139.91.255.0/24. Among these subnets, we would have chosen the top-two ones having the highest amount of traffic (first as sender, then as receiver). Indeed, one could imagine that Montefiore and RUN institute contribute to a very large percentage of the traffic volume inside the Uliège because of their networking applications.
This was have been done by using the aggregated tree from the previous question. Another method could have been to count, from all the networks with prefixes 16 and big traffics, the amount of subnetworks with prefix 14 that were inside. If two of those had also big traffic volumes, we would have found the 3 networks we are searching for. The functions for both approach were implemented and tested, however we were not able to determine a good guess for the Montefiore and RUN network addresses due to the size of the dataset and the computing time necessary to analyse it.

You can see in Table IV the overall traffic volume going in and out of the Uliege subnetwork. We note that there is the same magnitude of traffic sent and received at Uliege, which is to be expected as universities, in their Computer science departement, are doing among other things Network

measurements which rely on sending packets through the network of ASes.

| Uliege Received | Uliege Sent |
|---|---|
| 1451 GB | 851 GB |

TABLE IV: Traffic sent/received from/at Uliege. Extrapolation from 10% of the data.

| Montefiore Received | Montefiore Sent | RUN Received | RUN Sent |
|---|---|---|---|
| | | | |

TABLE V: Traffic sent/received from/at Montefiore Institute and RUN

## VII. Conclusion

In this project, we were able to learn a lot on network flow analysis and we drew a few conclusions. We found that the average packets size is 724 bytes, which is what we expected because it is half of the largest packet, an MTU packet of 1500 bytes.This is because there are as many small packets ([0; 200]) as there are large ones ([1200; 1400]).
Moreover, we found that the packets in the flows are mainly of size of the MTU packets. Analysing the port numbers, we noted that (without surprise) the most used ports are port 80 and 443 used by the http-service. Using the literature, we found an effective way to aggregate the traffic of different subnets by specifying the prefixes of the desired subnets.
We saw that anonymisation serves its purpose by allowing us to work with data and also to protect the privacy of the users. We were able to find, in the trace, the anonymized IP belonging to the Uliege subnetwork. Unfortunately, we were not able to find the network addresses of Montefiore and RUN subnets due to large datasets and therefore big computing times.

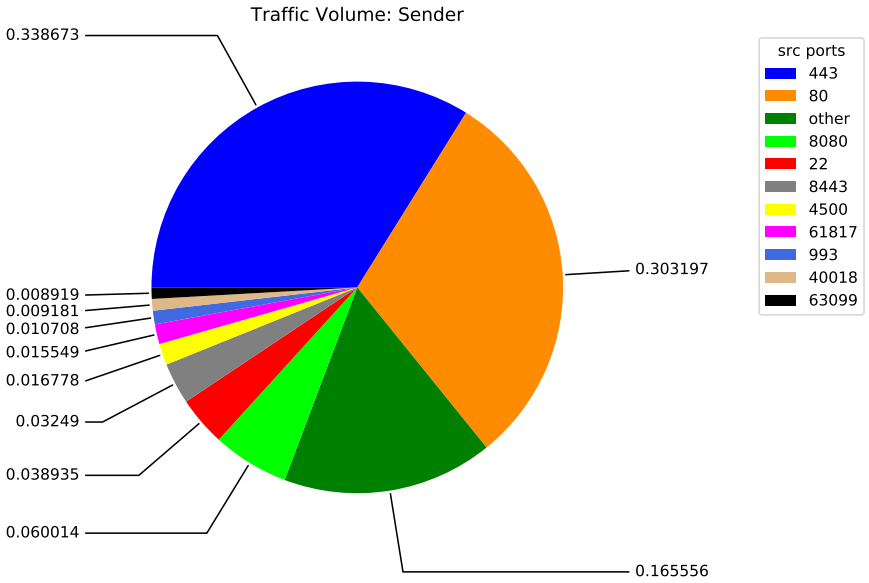



Fig. 7: Proportion of traffic volume per sender

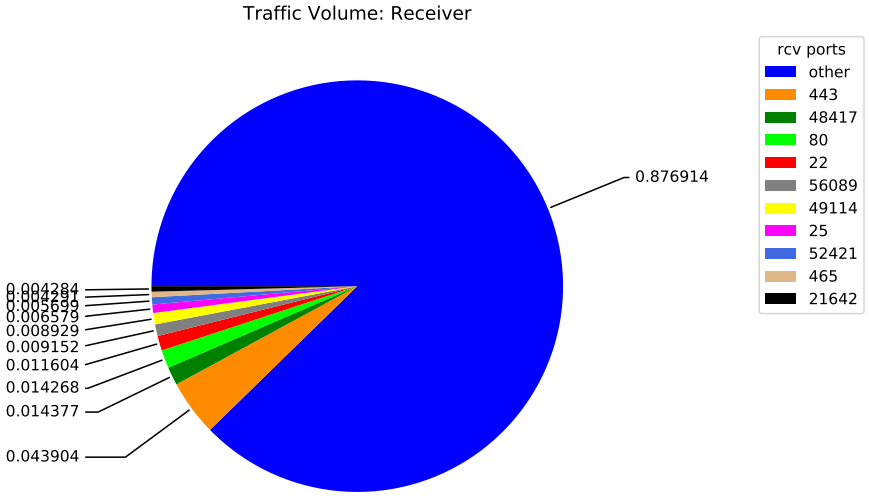



Fig. 8: Proportion of traffic volume per receiver

## REFERENCES

[1] G. V. Cristian Estan, Stefan Savage, "Automatically inferring patterns of resource consumption in network traffic," 2003.