

# assignment3

October 13, 2021

```
[1]: import networkx as nx
import matplotlib.pyplot as plt
import seaborn as sns
import csv
import numpy as np
```

```
[8]: import glob
files = glob.glob("exported-flows/*.csv")
#glob.glob("./exported-flows/*")
print(files)
```

```
['exported-flows/ft-v05.2016-05-26.210712+0200.csv', 'exported-
flows/ft-v05.2016-05-26.161912+0200.csv', 'exported-
flows/ft-v05.2016-05-26.230225+0200.csv', 'exported-
flows/ft-v05.2016-05-26.200936+0200.csv', 'exported-
flows/ft-v05.2016-05-26.191201+0200.csv', 'exported-
flows/ft-v05.2016-05-26.152136+0200.csv', 'exported-
flows/ft-v05.2016-05-26.220448+0200.csv', 'exported-
flows/ft-v05.2016-05-26.181424+0200.csv', 'exported-
flows/ft-v05.2016-05-26.171648+0200.csv', 'exported-
flows/ft-v05.2016-05-26.142400+0200.csv']
```

```
[24]: total_entries = 0
total_source_ips = 0
total_destination_ips = 0
total_packets_transferred = 0
list_of_protocols = []

set_s_ips = set()
set_d_ips = set()
G = nx.Graph()
```

```
[25]: for file in files:
    with open(file, "r") as opened_file:
        read_data = csv.reader(opened_file, delimiter=",", quotechar='"')
        next(read_data)
        for line in read_data:
            total_entries += 1
```

```
total_packets_transferred += int(line[4])

# ip statistics
source_ip = line[10]
dest_ip = line[11]
if source_ip not in set_s_ips:
    set_s_ips.add(source_ip)
if dest_ip not in set_d_ips:
    set_d_ips.add(dest_ip)

#protocol statistics
curr_prot = int(line[17])
list_of_protocols.append(curr_prot)

# only TCP, UDP
if curr_prot == 6 or curr_prot == 17:
    G.add_edge(source_ip, dest_ip)

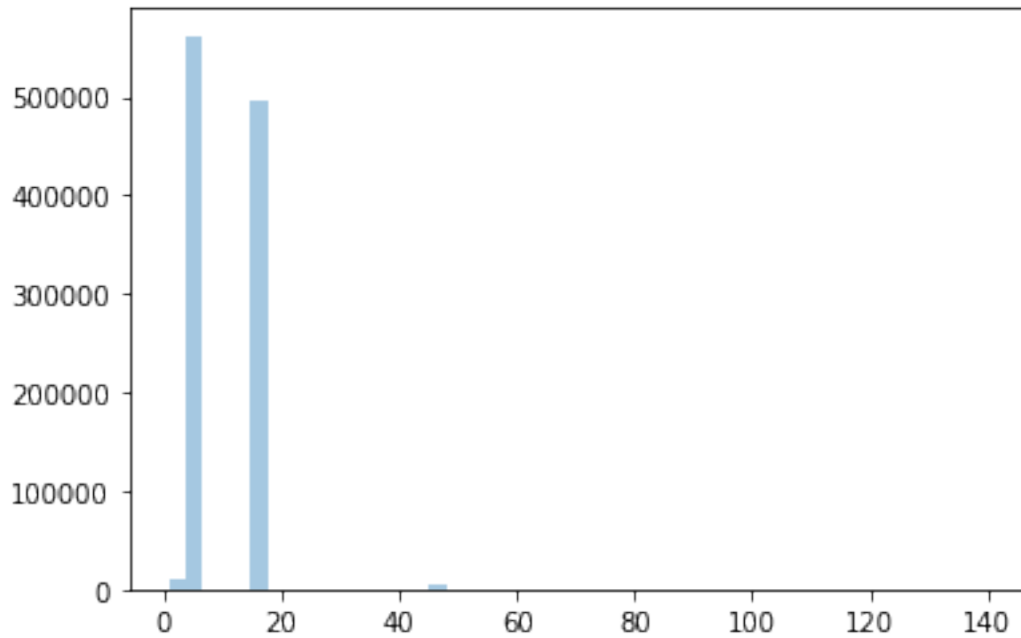
total_source_ips = len(set_s_ips)
```

```
[17]: total_source_ips
```

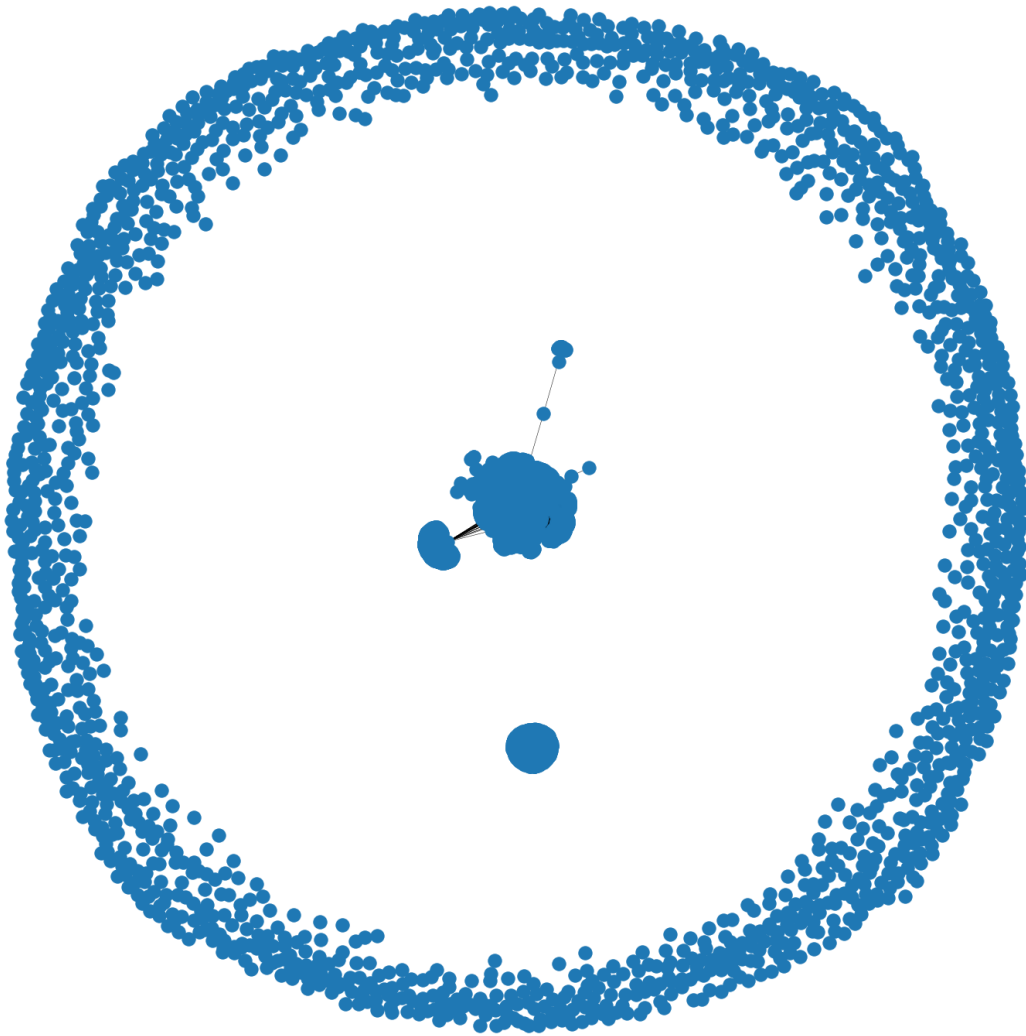
```
[17]: 7918
```

```
[23]: sns.distplot(list_of_protocols, kde=False)
      #plt.xlim([0, 20])
      #plt.xscale("log")
```

```
[23]: <AxesSubplot:>
```



```
[33]: plt.figure(figsize=(20, 20))
      from random import sample
      sample_nodes = sample(G.nodes(), 5000)
      G_sampled = G.subgraph(sample_nodes)
      pos = nx.spring_layout(G_sampled)
      nx.draw(G_sampled, width=0.5, pos=pos)
      #plt.savefig("g.png")
```



[ ]: