Student Name : Toh Yi Kang

Group : SCS1

Date : 28/03/2024

## LAB 4: ANALZING NETWORK DATA LOG

You are provided with the data file, in .csv format, in the working directory. Write the program to extract the following informations.

## EXERCISE 4A: TOP TALKERS AND LISTENERS

One of the most commonly used function in analyzing data log is finding out the IP address of the hosts that send out large amount of packet and hosts that receive large number of packets, usually know as TOP TALKERS and LISTENERS. Based on the IP address we can obtained the organization who owns the IP address.

List the TOP 5 TALKERS

| Rank | IP address | # of packets | Organisation |
|---|---|---|---|
| 1 | 193.62.192.8 | 3041 | European Bioinformatics Institute |
| 2 | 155.69.160.32 | 2975 | Nanyang Technological University |
| 3 | 130.14.250.11 | 2604 | National Library of Medicine |
| 4 | 14.139.196.58 | 2452 | Indian Institute of Technology |
| 5 | 140.112.8.139 | 2056 | AS17716 National Taiwan University |

TOP 5 LISTENERS

| Rank | IP address | # of packets | Organisation |
|---|---|---|---|
| 1 | 103.37.198.100 | 3841 | A*STAR |
| 2 | 137.132.228.15 | 3715 | National University of Singapore |
| 3 | 202.21.159.244 | 2446 | Rpnet |
| 4 | 192.101.107.153 | 2368 | Battelle Memorial Institute, Pacific Northwest Division |
| 5 | 103.21.126.2 | 2056 | Indian Institute of Technology Bombay |

## EXERCISE 4B: TRANSPORT PROTOCOL

Using the IP protocol type attribute, determine the percentage of TCP and UDP protocol

|  | Header value | Transport layer protocol | # of packets | % |
|---|---|---|---|---|
| 1 | 6 | TCP | 56064 | 80.818798 |
| 2 | 17 | UDP | 9462 | 13.639902 |

## EXERCISE 4C: APPLICATIONS PROTOCOL

Using the Destination IP port number determine the most frequently used application protocol. (For finding the service given the port number https://www.adminsub.net/tcp-udp-port-finder/ )

| Rank | Destination IP port number | # of packets | Service |
|---|---|---|---|
| 1 | 443 | 13423 | HTTPS |
| 2 | 80 | 2647 | HTTP |
| 3 | 52866 | 2068 | Dynamic/Private Ports |
| 4 | 45512 | 1356 | Unassigned |
| 5 | 56152 | 1341 | Dynamic/Private Ports |

## EXERCISE 4D: TRAFFIC

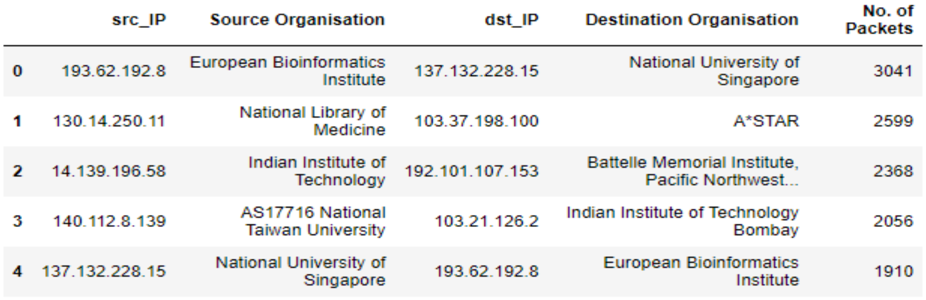
The traffic intensity is an important parameter that a network engineer needs to monitor closely to determine if there is congestion. You would use the IP packet size to calculate the estimated total traffic over the monitored period of 15 seconds. (Assume the sampling rate is 1 in 2048)
Mainly summing up all packet size (in bytes already) together, which is the total size for one sample. Total traffic wil be the (sum*2048 samples)/(2^20) – 2^20 conversion to convert to MB.

| | |
|---|---|
| Total Traffic( MB) | 126519.184 MB |

## EXERCISE 4E: ADDITIONAL ANALYSIS

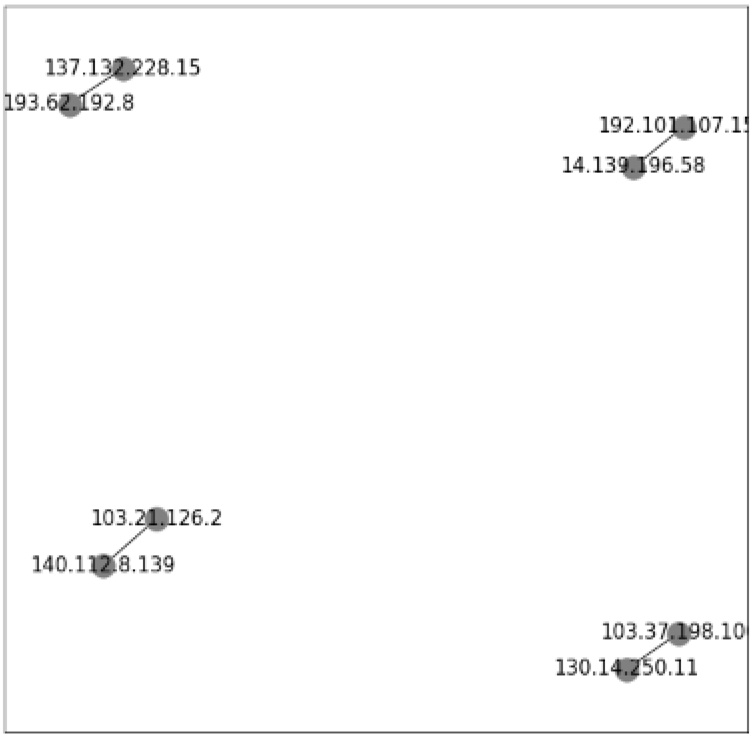
Please append ONE page to provide additional analysis of the data and the insight it provides. Please limit your results within one page (and any additional results that fall beyond one page limit will not be assessed).

Top 5 Communication Pairs based on number of packets between organizations:



| | src_IP | Source Organisation | dst_IP | Destination Organisation | No. of Packets |
|---|---|---|---|---|---|
| 0 | 193.62.192.8 | European Bioinformatics Institute | 137.132.228.15 | National University of Singapore | 3041 |
| 1 | 130.14.250.11 | National Library of Medicine | 103.37.198.100 | A*STAR | 2599 |
| 2 | 14.139.196.58 | Indian Institute of Technology | 192.101.107.153 | Battelle Memorial Institute, Pacific Northwest... | 2368 |
| 3 | 140.112.8.139 | AS17716 National Taiwan University | 103.21.126.2 | Indian Institute of Technology Bombay | 2056 |
| 4 | 137.132.228.15 | National University of Singapore | 193.62.192.8 | European Bioinformatics Institute | 1910 |

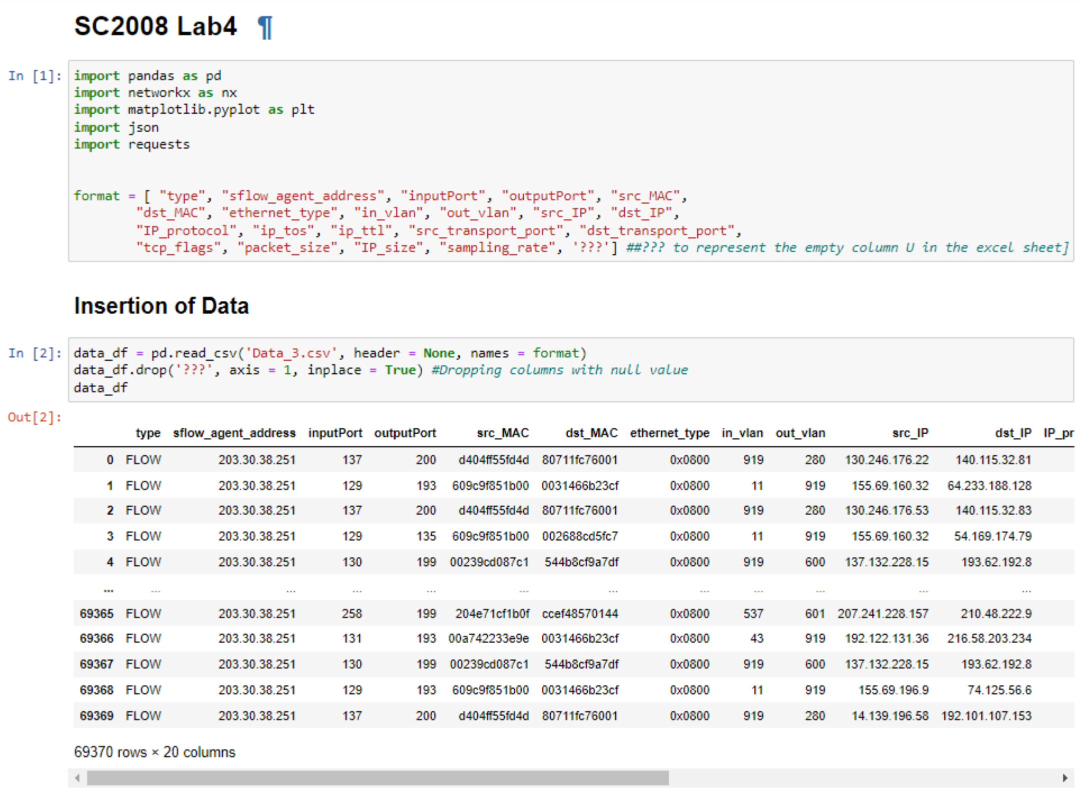
Visualization of top 5 communications pairs:



Although we state out the top 5 communication pairs, there are only 4 shown in the graph. This is mainly because the first and fifth communication paris have the same IP hosts but in different directions. Since the graph shown is non-directional, we can only display mainly first to fourth communication pairs since the fifth pair is technically same as the first pair.

## EXERCISE 4F: SOFTWARE CODE

Please also submit your code to the NTULearn lab site.



### SC2008 Lab4 ¶

In [1]:
```
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt
import json
import requests


format = [ "type", "sflow_agent_address", "inputPort", "outputPort", "src_MAC",
        "dst_MAC", "ethernet_type", "in_vlan", "out_vlan", "src_IP", "dst_IP",
        "IP_protocol", "ip_tos", "ip_ttl", "src_transport_port", "dst_transport_port",
        "tcp_flags", "packet_size", "IP_size", "sampling_rate", '???'] ##??? to represent the empty column U in the excel sheet]
```

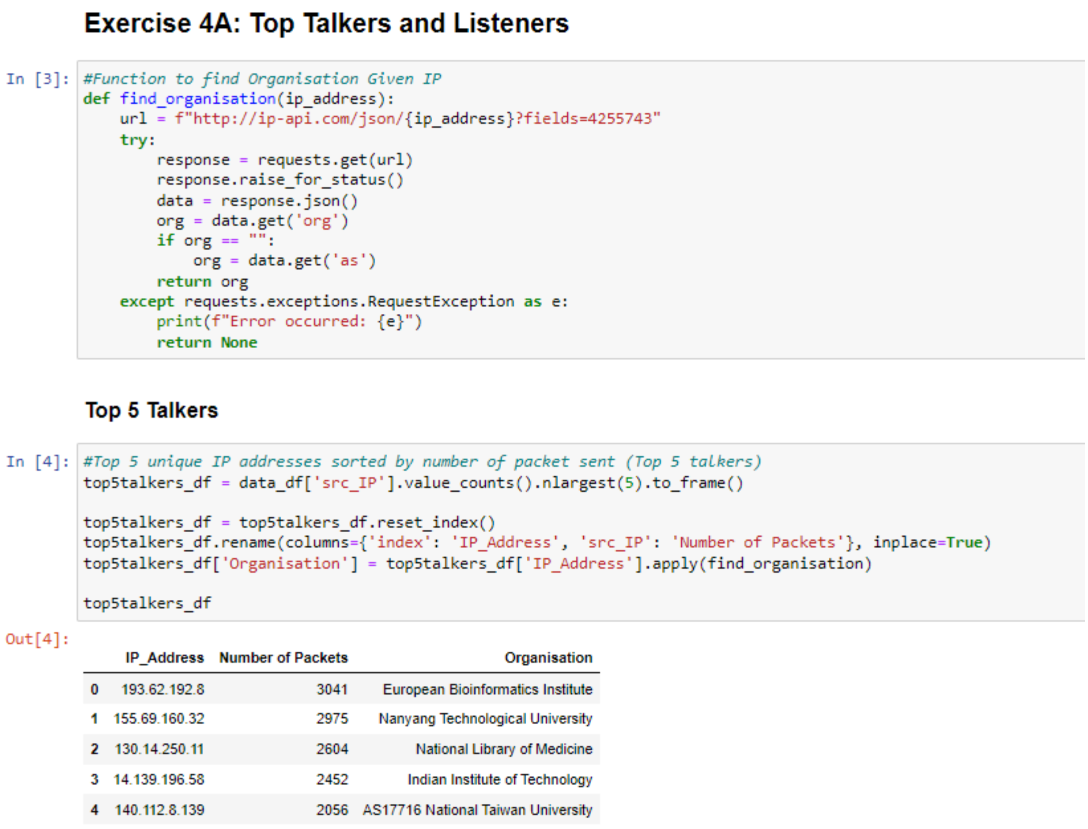
#### Insertion of Data

In [2]:
```
data_df = pd.read_csv('Data_3.csv', header = None, names = format)
data_df.drop('???', axis = 1, inplace = True) #Dropping columns with null value
data_df
```

Out[2]:

| | type | sflow_agent_address | inputPort | outputPort | src_MAC | dst_MAC | ethernet_type | in_vlan | out_vlan | src_IP | dst_IP | IP_pr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | FLOW | 203.30.38.251 | 137 | 200 | d404ff55fd4d | 80711fc76001 | 0x0800 | 919 | 280 | 130.246.176.22 | 140.115.32.81 | |
| 1 | FLOW | 203.30.38.251 | 129 | 193 | 609c9f851b00 | 0031466b23cf | 0x0800 | 11 | 919 | 155.69.160.32 | 64.233.188.128 | |
| 2 | FLOW | 203.30.38.251 | 137 | 200 | d404ff55fd4d | 80711fc76001 | 0x0800 | 919 | 280 | 130.246.176.53 | 140.115.32.83 | |
| 3 | FLOW | 203.30.38.251 | 129 | 135 | 609c9f851b00 | 002688cd5fc7 | 0x0800 | 11 | 919 | 155.69.160.32 | 54.169.174.79 | |
| 4 | FLOW | 203.30.38.251 | 130 | 199 | 00239cd087c1 | 544b8cf9a7df | 0x0800 | 919 | 600 | 137.132.228.15 | 193.62.192.8 | |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| 69365 | FLOW | 203.30.38.251 | 258 | 199 | 204e71cf1b0f | ccef48570144 | 0x0800 | 537 | 601 | 207.241.228.157 | 210.48.222.9 | |
| 69366 | FLOW | 203.30.38.251 | 131 | 193 | 00a742233e9e | 0031466b23cf | 0x0800 | 43 | 919 | 192.122.131.36 | 216.58.203.234 | |
| 69367 | FLOW | 203.30.38.251 | 130 | 199 | 00239cd087c1 | 544b8cf9a7df | 0x0800 | 919 | 600 | 137.132.228.15 | 193.62.192.8 | |
| 69368 | FLOW | 203.30.38.251 | 129 | 193 | 609c9f851b00 | 0031466b23cf | 0x0800 | 11 | 919 | 155.69.196.9 | 74.125.56.6 | |
| 69369 | FLOW | 203.30.38.251 | 137 | 200 | d404ff55fd4d | 80711fc76001 | 0x0800 | 919 | 280 | 14.139.196.58 | 192.101.107.153 | |

69370 rows × 20 columns



### Exercise 4A: Top Talkers and Listeners

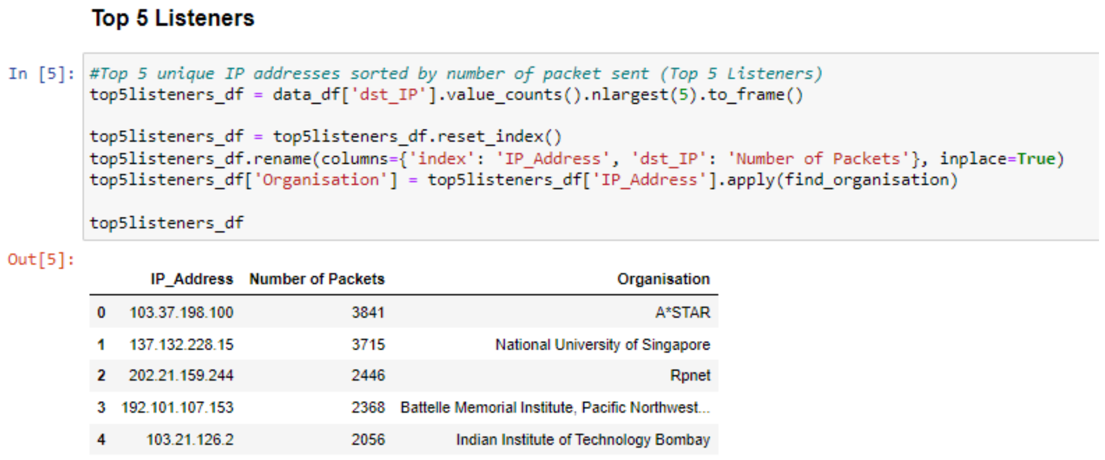
In [3]:
```
#Function to find Organisation Given IP
def find_organisation(ip_address):
    url = f"http://ip-api.com/json/{ip_address}?fields=4255743"
    try:
        response = requests.get(url)
        response.raise_for_status()
        data = response.json()
        org = data.get('org')
        if org == "":
            org = data.get('as')
        return org
    except requests.exceptions.RequestException as e:
        print(f"Error occurred: {e}")
        return None
```

#### Top 5 Talkers

In [4]:
```
#Top 5 unique IP addresses sorted by number of packet sent (Top 5 talkers)
top5talkers_df = data_df['src_IP'].value_counts().nlargest(5).to_frame()

top5talkers_df = top5talkers_df.reset_index()
top5talkers_df.rename(columns={'index': 'IP_Address', 'src_IP': 'Number of Packets'}, inplace=True)
top5talkers_df['Organisation'] = top5talkers_df['IP_Address'].apply(find_organisation)

top5talkers_df
```

Out[4]:

| | IP_Address | Number of Packets | Organisation |
|---|---|---|---|
| 0 | 193.62.192.8 | 3041 | European Bioinformatics Institute |
| 1 | 155.69.160.32 | 2975 | Nanyang Technological University |
| 2 | 130.14.250.11 | 2604 | National Library of Medicine |
| 3 | 14.139.196.58 | 2452 | Indian Institute of Technology |
| 4 | 140.112.8.139 | 2056 | AS17716 National Taiwan University |

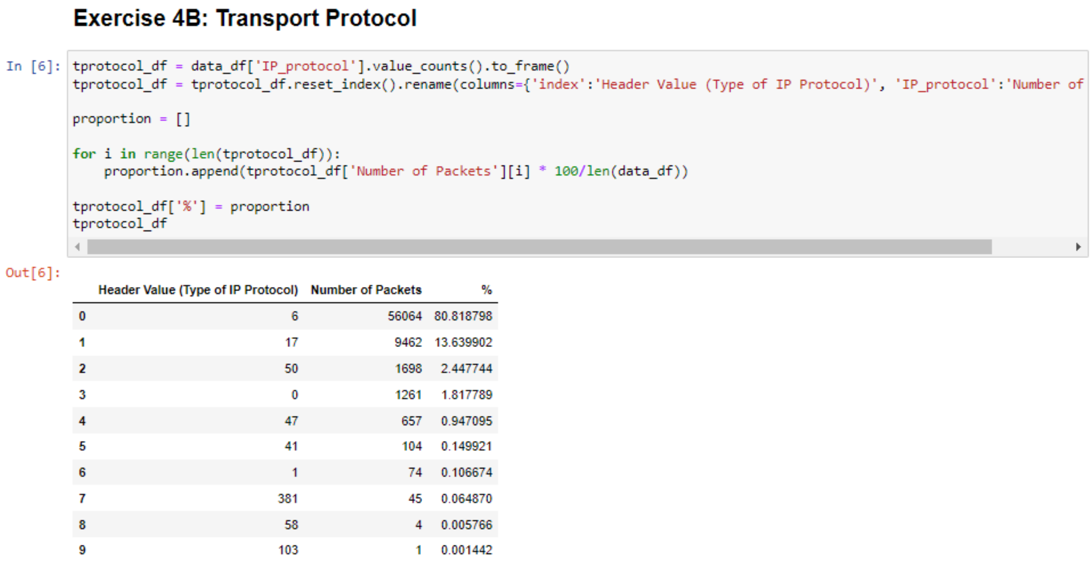


### Top 5 Listeners

In [5]:
```
#Top 5 unique IP addresses sorted by number of packet sent (Top 5 Listeners)
top5listeners_df = data_df['dst_IP'].value_counts().nlargest(5).to_frame()

top5listeners_df = top5listeners_df.reset_index()
top5listeners_df.rename(columns={'index': 'IP_Address', 'dst_IP': 'Number of Packets'}, inplace=True)
top5listeners_df['Organisation'] = top5listeners_df['IP_Address'].apply(find_organisation)

top5listeners_df
```

Out[5]:

| | IP_Address | Number of Packets | Organisation |
|---|---|---|---|
| 0 | 103.37.198.100 | 3841 | A*STAR |
| 1 | 137.132.228.15 | 3715 | National University of Singapore |
| 2 | 202.21.159.244 | 2446 | Rpnet |
| 3 | 192.101.107.153 | 2368 | Battelle Memorial Institute, Pacific Northwest... |
| 4 | 103.21.126.2 | 2056 | Indian Institute of Technology Bombay |

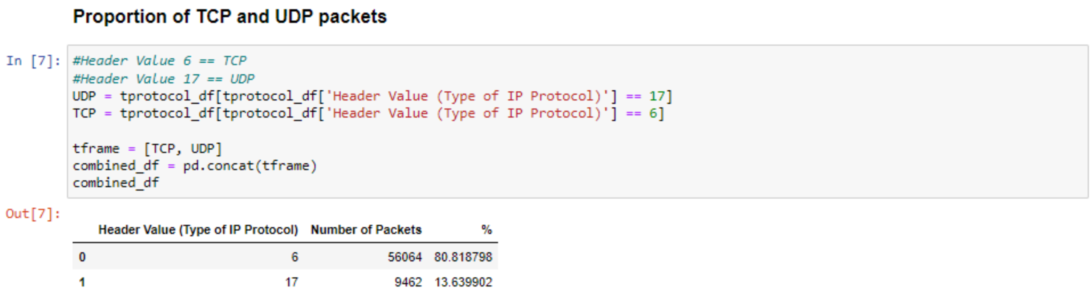


### Exercise 4B: Transport Protocol

In [6]:
```
tprotocol_df = data_df['IP_protocol'].value_counts().to_frame()
tprotocol_df = tprotocol_df.reset_index().rename(columns={'index':'Header Value (Type of IP Protocol)', 'IP_protocol':'Number of

proportion = []

for i in range(len(tprotocol_df)):
    proportion.append(tprotocol_df['Number of Packets'][i] * 100/len(data_df))

tprotocol_df['%'] = proportion
tprotocol_df
```

Out[6]:

| | Header Value (Type of IP Protocol) | Number of Packets | % |
|---|---|---|---|
| 0 | 6 | 56064 | 80.818798 |
| 1 | 17 | 9462 | 13.639902 |
| 2 | 50 | 1698 | 2.447744 |
| 3 | 0 | 1261 | 1.817789 |
| 4 | 47 | 657 | 0.947095 |
| 5 | 41 | 104 | 0.149921 |
| 6 | 1 | 74 | 0.106674 |
| 7 | 381 | 45 | 0.064870 |
| 8 | 58 | 4 | 0.005766 |
| 9 | 103 | 1 | 0.001442 |

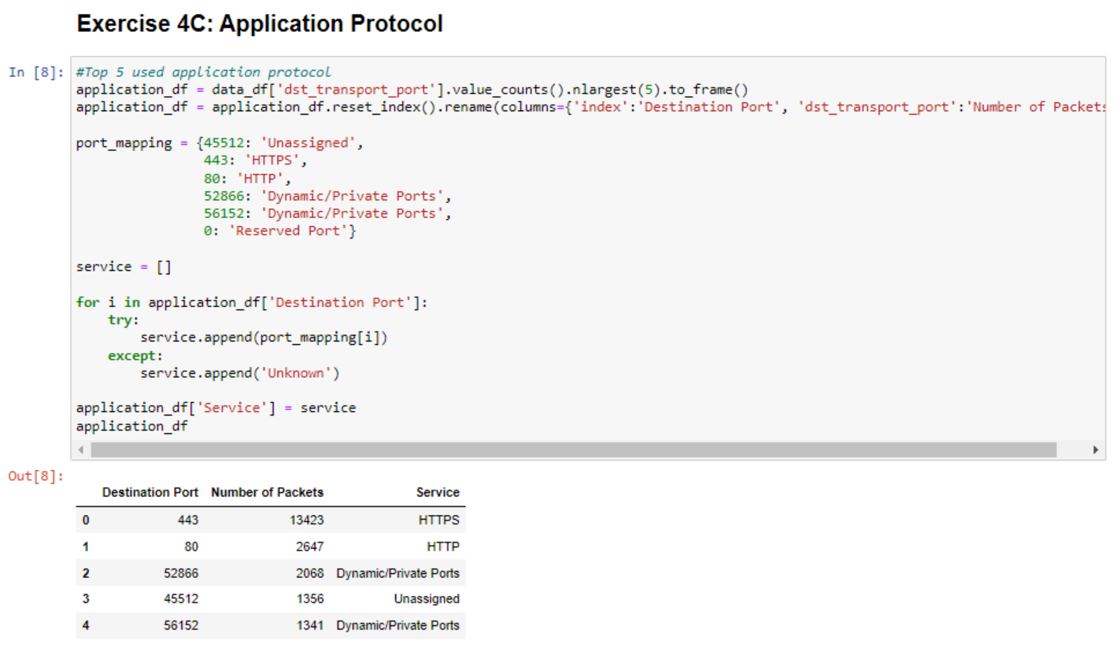


### Proportion of TCP and UDP packets

In [7]:
```
#Header Value 6 == TCP
#Header Value 17 == UDP
UDP = tprotocol_df[tprotocol_df['Header Value (Type of IP Protocol)'] == 17]
TCP = tprotocol_df[tprotocol_df['Header Value (Type of IP Protocol)'] == 6]

tframe = [TCP, UDP]
combined_df = pd.concat(tframe)
combined_df
```

Out[7]:

| | Header Value (Type of IP Protocol) | Number of Packets | % |
|---|---|---|---|
| 0 | 6 | 56064 | 80.818798 |
| 1 | 17 | 9462 | 13.639902 |

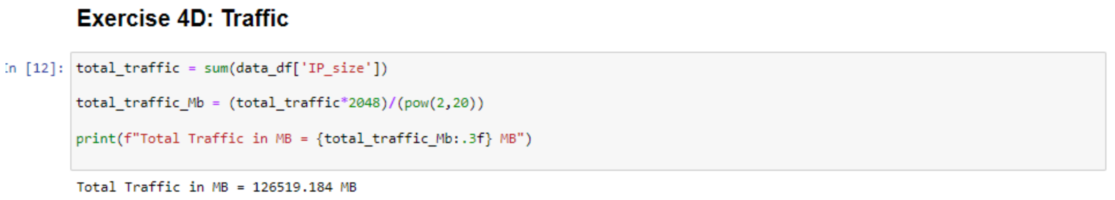


## Exercise 4C: Application Protocol

```
In [8]:
#Top 5 used application protocol
application_df = data_df['dst_transport_port'].value_counts().nlargest(5).to_frame()
application_df = application_df.reset_index().rename(columns={'index':'Destination Port', 'dst_transport_port':'Number of Packets

port_mapping = {45512: 'Unassigned',
                443: 'HTTPS',
                80: 'HTTP',
                52866: 'Dynamic/Private Ports',
                56152: 'Dynamic/Private Ports',
                0: 'Reserved Port'}

service = []

for i in application_df['Destination Port']:
    try:
        service.append(port_mapping[i])
    except:
        service.append('Unknown')

application_df['Service'] = service
application_df
```

Out[8]:

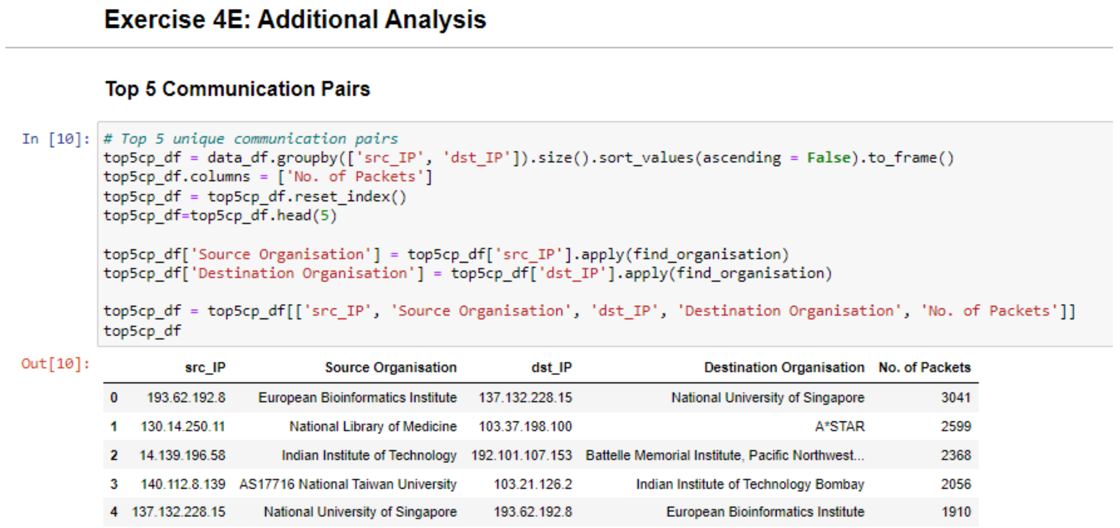
| | Destination Port | Number of Packets | Service |
|---|---|---|---|
| 0 | 443 | 13423 | HTTPS |
| 1 | 80 | 2647 | HTTP |
| 2 | 52866 | 2068 | Dynamic/Private Ports |
| 3 | 45512 | 1356 | Unassigned |
| 4 | 56152 | 1341 | Dynamic/Private Ports |



## Exercise 4D: Traffic

```
:n [12]:
total_traffic = sum(data_df['IP_size'])

total_traffic_Mb = (total_traffic*2048)/(pow(2,20))

print(f"Total Traffic in MB = {total_traffic_Mb:.3f} MB")
```

```
Total Traffic in MB = 126519.184 MB
```



## Exercise 4E: Additional Analysis

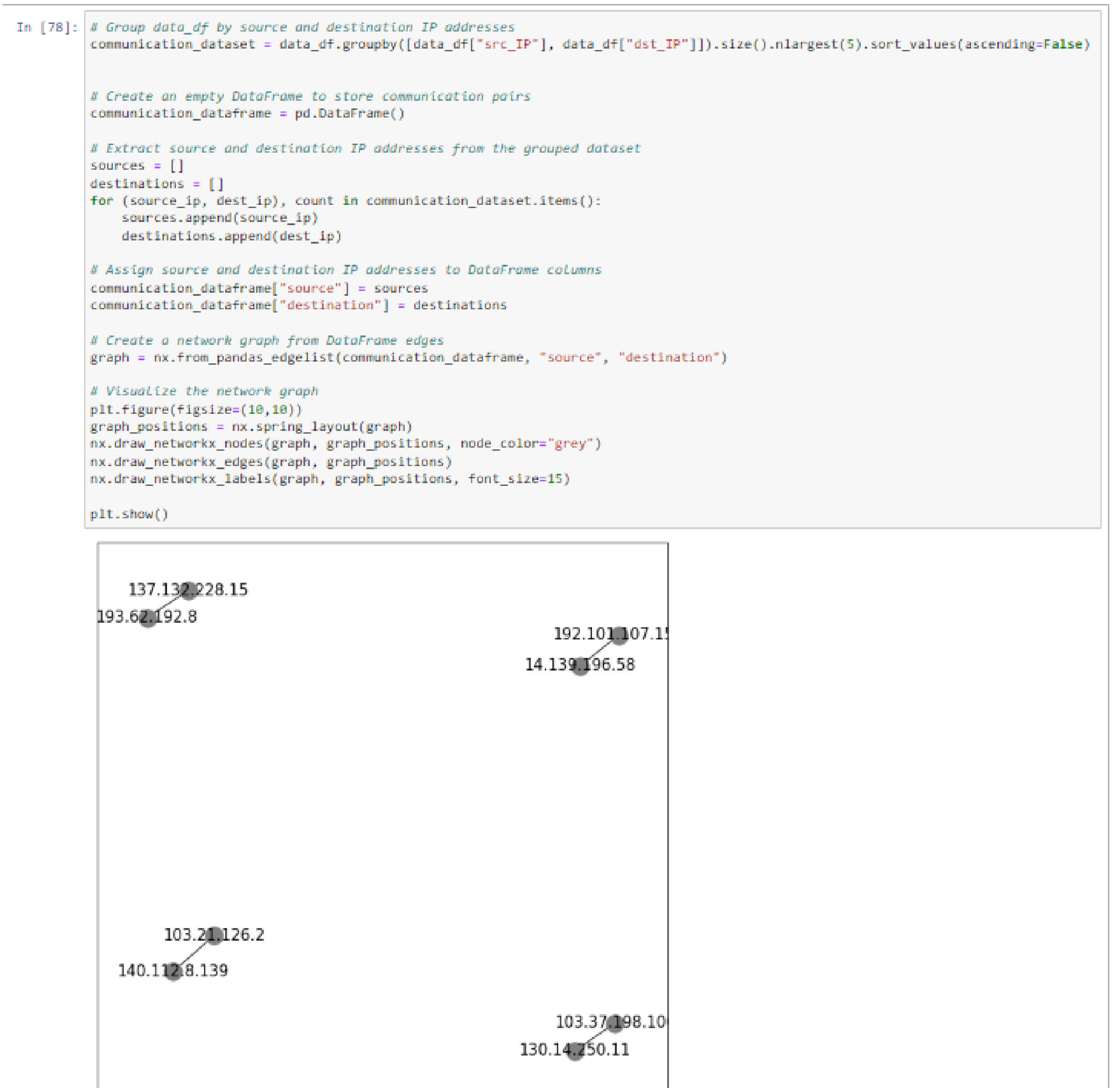
### Top 5 Communication Pairs

```
In [10]:
# Top 5 unique communication pairs
top5cp_df = data_df.groupby(['src_IP', 'dst_IP']).size().sort_values(ascending = False).to_frame()
top5cp_df.columns = ['No. of Packets']
top5cp_df = top5cp_df.reset_index()
top5cp_df=top5cp_df.head(5)

top5cp_df['Source Organisation'] = top5cp_df['src_IP'].apply(find_organisation)
top5cp_df['Destination Organisation'] = top5cp_df['dst_IP'].apply(find_organisation)

top5cp_df = top5cp_df[['src_IP', 'Source Organisation', 'dst_IP', 'Destination Organisation', 'No. of Packets']]
top5cp_df
```

Out[10]:

| | src_IP | Source Organisation | dst_IP | Destination Organisation | No. of Packets |
|---|---|---|---|---|---|
| 0 | 193.62.192.8 | European Bioinformatics Institute | 137.132.228.15 | National University of Singapore | 3041 |
| 1 | 130.14.250.11 | National Library of Medicine | 103.37.198.100 | A*STAR | 2599 |
| 2 | 14.139.196.58 | Indian Institute of Technology | 192.101.107.153 | Battelle Memorial Institute, Pacific Northwest... | 2368 |
| 3 | 140.112.8.139 | AS17716 National Taiwan University | 103.21.126.2 | Indian Institute of Technology Bombay | 2056 |
| 4 | 137.132.228.15 | National University of Singapore | 193.62.192.8 | European Bioinformatics Institute | 1910 |

### Visualization of top 5 communications pairs between different IP hosts



In [78]:

```
# Group data_df by source and destination IP addresses
communication_dataset = data_df.groupby([data_df["src_IP"], data_df["dst_IP"]]).size().nlargest(5).sort_values(ascending=False)


# Create an empty DataFrame to store communication pairs
communication_dataframe = pd.DataFrame()

# Extract source and destination IP addresses from the grouped dataset
sources = []
destinations = []
for (source_ip, dest_ip), count in communication_dataset.items():
    sources.append(source_ip)
    destinations.append(dest_ip)

# Assign source and destination IP addresses to DataFrame columns
communication_dataframe["source"] = sources
communication_dataframe["destination"] = destinations

# Create a network graph from DataFrame edges
graph = nx.from_pandas_edgelist(communication_dataframe, "source", "destination")

# Visualize the network graph
plt.figure(figsize=(10,10))
graph_positions = nx.spring_layout(graph)
nx.draw_networkx_nodes(graph, graph_positions, node_color="grey")
nx.draw_networkx_edges(graph, graph_positions)
nx.draw_networkx_labels(graph, graph_positions, font_size=15)

plt.show()
```