Student Name : Jervis Chan Jun Yong

Group : TS8

Date : 7 April 2021

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 (NLM)
4	14.139.196.58	2452	Indian Institute of Technology (IIT), Guwahati
5	140.112.8.139	2056	Taiwan Academic Network, Ministry of Education computer Center

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	Republic Polytechnic
4	192.101.107.153	2368	Pacific Northwest National Laboratory (PNNL-Z)
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 (82.37%)
2	17	UDP	9462 (13.90%)
		All Protocols	68065

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 Port and/or Private Ports Xsan. Xsan Filesystem Access (Apple)
4	45512	1356	Unassigned Port
5	56152	1341	Dynamic Port and/or Private Ports Xsan. Xsan Filesystem Access (Apple)

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 1000)

Total Traffic(MB)	Assuming sampling rate of 1 in 1000 64777.822MB (base10 MB to B conversion) 61776.945 (base2 MB to B conversion)

EXERCISE 4E: ADDITIONAL ANALYSIS

EXERCISE 4E(A): TOP 5 COMMUNICATION PAIRS

pair_a	a_org	pair_b	b_org	count
137.132.228.15	National University of Singapore	193.62.192.8	Janet University Network	4951
103.37.198.100	A-STAR	130.14.250.11	National Library of Medicine	2842
14.139.196.58	NKN EDGE Network	192.101.107.153	ESnet	2368
103.21.126.2	Indian Institute of Technology Bombay	140.112.8.139	National Taiwan University	2056
140.90.101.61	Noaa-silverspring	167.205.52.8	Institute of Technology Bandung	1752

EXERCISE 4E(B): VISUALISATION OF TOP TALKER/LISTENER COMMUNICATIONS

Out of curiosity, I would love to know the kind of communications of the top 5 listeners and top 5 talkers. Are they exclusively communicating with one IP? Or are they communicating with many different IPs?

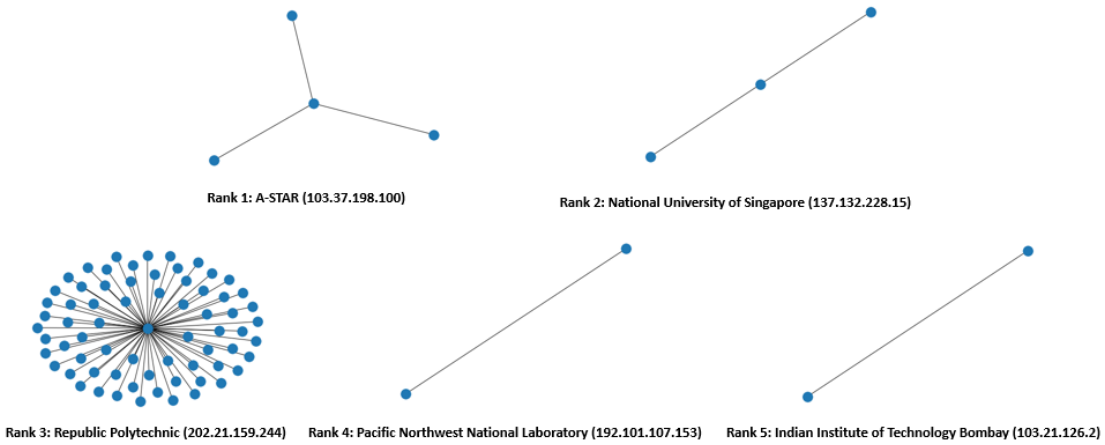

Figure 1: Visualisation of Top 5 Listeners

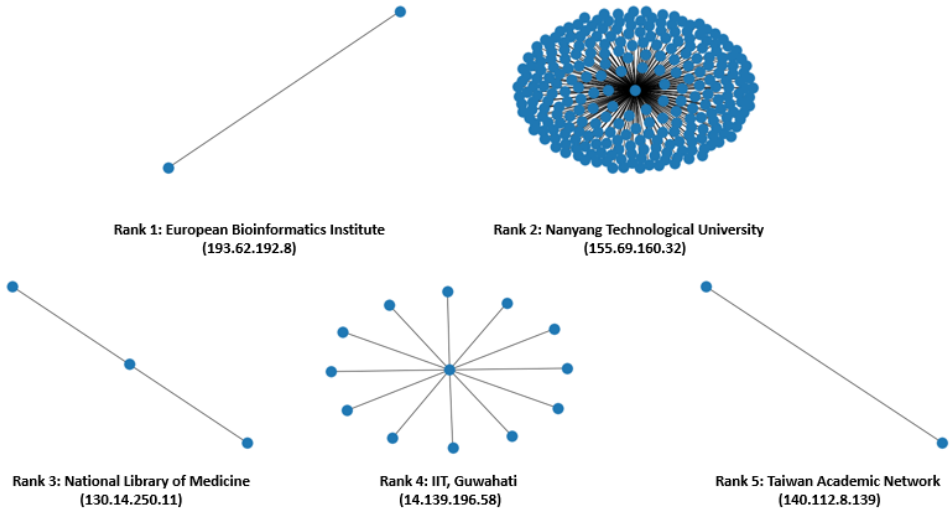

Figure 2: Visualisation of Top 5 Talkers

After visualizing, it could be more likely that the data scope is limited, leading to more data from NTU.

EXERCISE 4F: SOFTWARE CODE

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

In [ ]:
```
SFlow_data = pd.read_csv("SFlow_Data_lab4.csv", header = None)
SFlow_data.columns = ["Type", "sflow_agent_address", "inputPort", "outputPort", "src_MAC", "dst_MAC", "ethernet_type",
SFlow_data.head()
```

Top Talkers and Listeners

In [ ]:
```
top_5_talkers = SFlow_data.pivot_table(index=['src_IP'], aggfunc='size').sort_values(ascending=False).iloc[0:5]
top_5_talkers
```

In [ ]:
```
top_5_listeners = SFlow_data.pivot_table(index=['dst_IP'], aggfunc='size').sort_values(ascending=False).iloc[0:5]
top_5_listeners
```

Transport Protocol

In [ ]:
```
top_5_packets = SFlow_data.pivot_table(index=['IP_protocol'], aggfunc='size').sort_values(ascending=False).iloc[0:5]
total_packets = top_5_packets.sum()
print(top_5_packets)
```

In [ ]:
```
top_5_packets /= total_packets
top_5_packets
```

Applications Protocol

In [ ]:
```
top_5_packets = SFlow_data.pivot_table(index=['udp_dst_port/tcp_dst_port/icmp_code'], aggfunc='size').sort_values(asc
top_5_packets
```

Traffic

In [ ]:
```
sampling_rates = SFlow_data.pivot_table(index=['sampling_rate'], aggfunc='size').sort_values(ascending=False)
sampling_rates
```

In [ ]:
```
# Base10 MB to B Conversion
traffic = SFlow_data["IP_size"] * 1000
total_traffic = traffic.sum() / 10**6
print("Total Traffic (1): ", total_traffic, "MB")

# Base2 MB to B Conversion
traffic = SFlow_data["IP_size"] * 1000
total_traffic = traffic.sum() / 2**20
print("Total Traffic (2): ", total_traffic, "MB")
```

Additional Analysis

Top 5 Communication Pairs

In [ ]:
```
pair_data = SFlow_data
pair_data['pair_IP'] = None

for index, row in pair_data.iterrows():
    pair = [row['src_IP'], row['dst_IP']]
    pair.sort()
    pair_ = tuple(pair)
    pair_data.at[index, 'pair_IP'] = pair_

pair_data_grouped = pair_data.groupby('pair_IP').size().reset_index(name='Count')
pair_data_grouped = pair_data_grouped.sort_values(['Count'], ascending=False)
top_5_pairs = pair_data_grouped.head(5)
top_5_pairs
```

Visualisations of Top Listeners

In [ ]:
```
for listener in top_5_listeners.keys():
    listener_network = SFlow_data.loc[SFlow_data['dst_IP'] == listener]
    listener_network = listener_network[['src_IP', 'dst_IP']]
    print("dst_IP:", listener)

    G = nx.Graph()
    G = nx.from_pandas_edgelist(listener_network, 'src_IP', 'dst_IP')
    print(list(G.nodes))
    nx.draw(G, with_labels=False)
    plt.show()
```

Visualisations of Top Talkers

In [ ]:
```
for talker in top_5_talkers.keys():
    talker_network = SFlow_data.loc[SFlow_data['src_IP'] == talker]
    talker_network = talker_network[['src_IP', 'dst_IP']]
    print("src_IP:", talker)

    G = nx.Graph()
    G = nx.from_pandas_edgelist(talker_network, 'src_IP', 'dst_IP')
    print(list(G.nodes))
    nx.draw(G, with_labels=False)
    plt.show()
```