Student Name : Bryan Lim Cheng Yee

Group : TS4

Date : 16 April 2021

LAB 4: ANALYZING 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	Taiwan Academic Network

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, Singapore
4	192.101.107.153	2368	Pacific Northwest National Laboratory
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%)
3	50	ESP	1698 (2.49%)
4	47	GRE	657 (0.97%)

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 port
4	45512	1356	dynamic/private port
5	56152	1341	dynamic/private port

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)	61776.945

EXERCISE 4E: ADDITIONAL ANALYSIS

Please append ONE page to provide additional analysis of the data and the insight it provides.
Examples include:
Top 5 communication pairs;
Visualization of communications between different IP hosts;
etc.

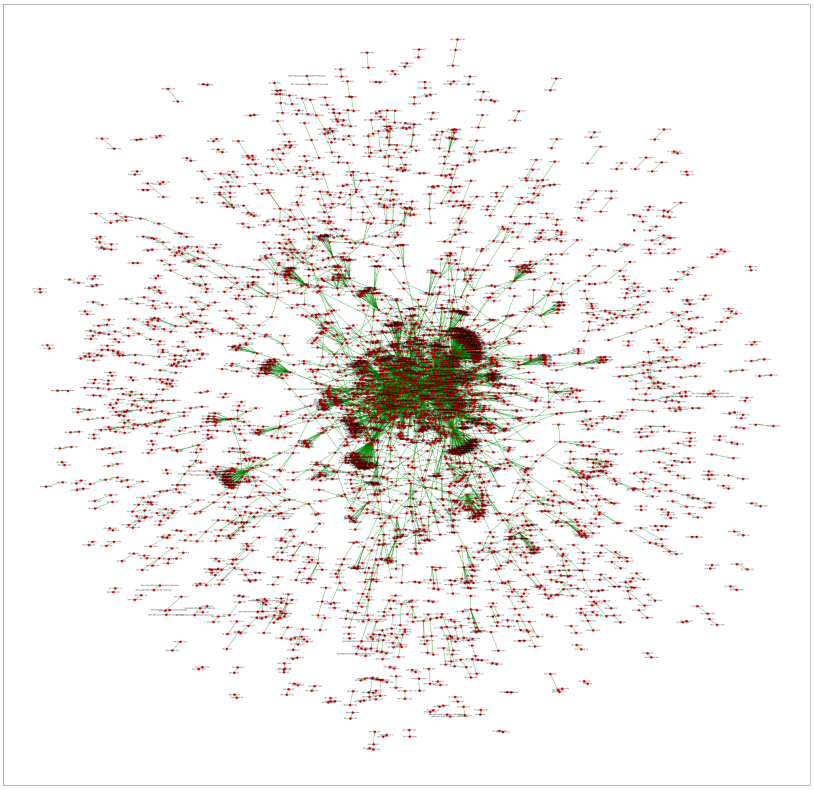
Please limit your results within one page (and any additional results that fall beyond one page limit will not be assessed).

Top 5 Communication Pairs

Rank	IP address 1	Organization 1	IP address 1	Organization 2	Count
1	137.132.228.15	National University of Singapore	193.62.192.8	European Bioinformatics Institute	4951
2	103.37.198.100	A*STAR	130.14.250.11	National Library of Medicine	2842
3	14.139.196.58	Indian Institute of Technology	192.101.107.153	Pacific Northwest National Library	2368
4	103.21.126.2	Powai	140.112.8.139	Taiwan Academic Network	2056
5	140.90.101.61	National Oceanic and Atmospheric Administration	167.205.52.8	Institut Teknologi Bandung	1752

It can be inferred from the data that the top communication pairs are usually between educational organizations, and research / data facilities.

Graph Showing Communication Links Between IP Hosts



From the figure, it can be inferred that out of all the IP hosts, there are a significant number of them who are linked very densely to each other, with multiple links between them.

EXERCISE 4F: SOFTWARE CODE

Please attach a copy of your code in an appendix.

Code:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import networkx as nx
import numpy as np
import requests
import math
import os

#read information from the csv file and display headers
network_data = "SFlow_Data_lab4.csv.csv"

df_raw = pd.read_csv(network_data, index_col = False, names=["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_port", "dst_port",
"tcp_flags", "packet_size", "ip_size", "sampling_rate", "others"])

df = df_raw[df_raw['Type']=='FLOW']

df.head(10)


#Exercise 4A-----------------------------------------

#List the top 5 talkers
top_5_talkers = df['src_IP'].value_counts()[:5]

print("{:<17}{:<9}".format("Sender IP", "Count"))
print(top_5_talkers)

print("-----------------------------------------")
print("The top 5 talkers are: ")
print('193.62.192.8 - European Bioinformatics Institute')
print('155.69.160.32 - Nanyang Technological University')
print('130.14.250.11 - National Library of Medicine')
print('14.139.196.58 - Indian Institute of Technology')
print('140.112.8.139 - Taiwan Academic Network')


#List the top 5 listeners
top_5_listeners = df['dst_IP'].value_counts()[:5]

print("{:<17}{:<9}".format("Listener IP", "Count"))
print(top_5_listeners)
```

```
#103.37.198.100
#137.132.228.15
#202.21.159.244
#192.101.107.153
#103.21.126.2

print("-------------------------------------------")
print("The top 5 listeners are: ")
print('103.37.198.100 - A*STAR ')
print('137.132.228.15 - National University of Singapore ')
print('202.21.159.244 - Republic Polytechnic, Singapore')
print('192.101.107.153 - Pacific Northwest National Laboratory ')
print('103.21.126.2 - Indian Institute of Technology Bombay')


#Exercise 4B-----------------------------------------

#Using the IP protocol type attribute,
#determine the percentage of TCP and UDP protocol
#TCP=6
#UDP=17
protocol_data = df['IP_protocol'].value_counts()

total_entries = len(df)

#print("{:<17}{:<9}".format("IP Protocol", "Count"))


print("{:<15}{:<9}{:<6}".format("Header       Value",       "Count",
"Proportion"))
for protocol in protocol_data.keys():
                         print("{:<15}{:<9}{:<6.2%}".format(protocol,
protocol_data[protocol], protocol_data[protocol]/total_entries))


print("-------------------------------------------")
print("Percentage of each protocol: ")
print('TCP - 82.37%')
print('UDP - 13.90%')
print('ESP - 2.49%')


#Exercise 4C-----------------------------------------

#Using the Destination IP port number determine the
#most frequently used application protocol.
#List the top 5 listeners
top_5_dst_port = df['dst_port'].value_counts()[:5]


print("{:<20}{:<9}".format("Dst IP Port No.", "Count"))
for app in top_5_dst_port.keys():
    print("{:<20}{:<9}".format(app, top_5_dst_port[app]))
```

```
#443 = https
#80 = http
#52866 = tcp/udp
#45512 = tcp/udp
#56152 = tcp/udp

print("-----------------------------------------")
print("The most frequently used application protocols are: ")
print('443 = https')
print('80 = http')
print('52866 = dynamic / private ports')
print('45512 = dynamic / private ports')
print('56152 = dynamic / private ports')


#Exercise 4D-----------------------------------------

#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)

df_packets = df

total_size_bytes = df_packets["ip_size"].sum()

total_size_mbytes = total_size_bytes / 1024 /1024
print("Traffic: {} MB". format(total_size_mbytes))

total_traffic = total_size_mbytes * 1000
print("Total traffic: {} MB". format(total_traffic))


#Exercise 4E-----------------------------------------

#Additional Analysis

#Top 5 communication pairs
df_comm_pairs = df
df_comm_pairs['ip_pair'] = None


for index, row in df_comm_pairs.iterrows():
    ip_pair_list = []
    ip_pair_list.append(row['dst_IP'])
    ip_pair_list.append(row['src_IP'])
    ip_pair_list.sort()
    ip_pair_tuple = tuple(ip_pair_list)
    df_comm_pairs.at[index, 'ip_pair'] = ip_pair_tuple

df_sections = df_comm_pairs.groupby('ip_pair')
df_section_count=df_sections.size().reset_index().rename(columns={0:'
count'})
```

```
df_sections=df_section_count.sort_values(['count'], ascending=False)

print("Top 5 communication pairs")
print(df_sections.head()[:5])


#Visualization of communications between different IP hosts
df_pair=dict(df.groupby(['src_IP',
'dst_IP']).size().sort_values(ascending = False))
df_pair = list(df_pair.items())
df_pair = [x[0] for x in df_pair]

graph = nx.Graph(df_pair)

layout = nx.spring_layout(graph, scale= 1.0, iterations = 20)
fig, ax = plt.subplots(1, 1, figsize = (50, 50))

print("Communication links between IP Hosts")
nx.draw_networkx(graph, pos = layout, with_labels=True, node_size =
60, font_size = 6, node_color='red', edge_color='green')
plt.show()


graph_pair = nx.path_graph(df_pair)
list(graph_pair.nodes)
```