Student Name: ____Lim Xin Yi________________________

Group : _____TCCA_______________________

Date : ______30/10/2023______________________

LAB 4: ANALZING NETWORK DATA LOG

Please download the data file, in .csv format, from the NTULearn lab site (not the lecture site). Write the program to do the exercises below.

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, using https://whatismyipaddress.com/

List the TOP 5 TALKERS

Rank	IP address	# of packets	Organisation
1	193.62.192.8	3041	European Bioinformatics
2	155.69.160.32	2975	NTU
3	130.14.250.11	2604	National Library of Medicine
4	14.139.196.58	2452	Indian Institute of Technology (IIT) Guwahati
5	140.112.8.139	2056	Taiwan Academic Network

List the TOP 5 LISTENERS

Rank	IP address	# of packets	Organisation
1	103.37.198.100	3841	A*STAR
2	137.132.228.15	3715	NUS
3	202.21.159.244	2446	Republic Polytechnic
4	192.101.107.153	2368	Battelle Memorial Institute Pacific Northwest Division
5	103.21.126.2	2056	Powai

EXERCISE 4B: TRANSPORT PROTOCOL

Using the IP_protocol field value, determine the number of packets for each protocol. For finding the transport layer protocol based on the IP_protocol field value, you can use https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml

	IP_protocol field value (in decimal number)	Transport layer protocol	# of packets
1	6	Transmission Control (TCP)	56064
2	17	User Datagram (UDP)	9462
3	50	Encap Security Payload	1698
4	47	Generic Routing Encapsulation	657
5	41	IL Transport Protocol	104
6	1	Internet Control Message (ICMP)	74
7	58	ICMP for IPv6 (IPv6-ICMP)	4
8	0	IPv6 Hop-by-Hop Option	1
9	103	Protocol Independent Multicast	1

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 and/or Private Ports
4	45512	1356	Unassigned
5	56152	1341	Dynamic and/or 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. (The sampling rate is 1 in 2048. You can either use 1MB = 2^{20} bytes or 1MB = 10^6 bytes. Both ways will be counted as correct.)

Total Traffic (MB)	64777822 bytes * 2048 / 10^6 = 132665 MB (6s.f.)

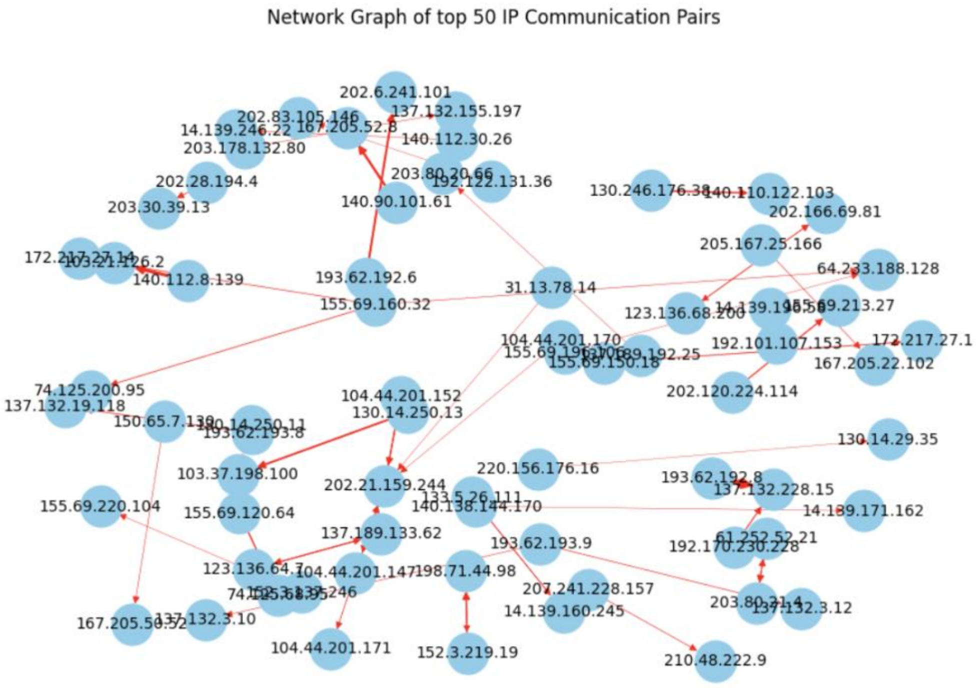
ANSWERS FOR PART 4(E) AND 4(F) ON NEXT PAGE

EXERCISE 4E: ADDITIONAL ANALYSIS

Top 5 communication pairs

src_IP	dst_IP	Count
193.62.192.8	**137.132.228.15**	3041
130.14.250.11	103.37.198.100	2599
14.139.196.58	192.101.107.153	2368
140.112.8.139	103.21.126.2	2056
137.132.228.15	**193.62.192.8**	1910

From the bolded rows, we can see that the top 1st and 5th communication pairs consist of the same two IP addresses 193.62.192.8 (European Bioinformatics) and 137.132.228.15 (NUS). This suggests that the communication between the two organisations contain request packets that result in responses sent. We also note that European Bioinformatics is the top talker while NUS is the second top listener as stated in part (A).



It is also observed that the network of top 50 IP communication pairs has an average node degree of 1.43, meaning each machine on average directly communicates with 1.43 other machines.

Protocol

ethernet_type	Protocol	Count
0x0800	Internet Protocol version 4 (IPv4)	67955
0x86dd	Internet Protocol version 6 (IPv6)	107
0x0806	Address Resolution Protocol (ARP)	2
0x0000		1

Almost 98% (67955 out of 68065) of all FLOW-type packets are IPv4 packets. There are 107 IPv6 packets, and 2 ARP packets.

EXERCISE 4F: SOFTWARE CODE

Please submit your code together with your answer sheet to the NTULearn lab site at the end of the laboratory session.

```
import pandas as pd
import matplotlib.pyplot as plt
import networkx as nx

df = pd.read_csv('SFlow_Data_lab4.csv', header=None)
df = df.iloc[:, :-1]
df.columns = ['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',
'udp_src_port/tcp_src_port/icmp_type', 'udp_dst_port/tcp_dst_port/icmp_code', 'tcp_flags',
'packet_size', 'IP_packet_size', 'sampling_rate']
df = df[df['type'] != 'CNTR']

def top_n(column_name, n):
  counts = df.groupby(column_name).size().reset_index(name='count')
  descending_counts = counts.sort_values(by='count', ascending=False)
  return descending_counts.head(n)



# part a
# top_5_talkers
print("\nTop 5 Talkers: ")
print(top_n('src_IP', 5))
# top_5_listeners
print("\nTop 5 Listeners: ")
print(top_n('dst_IP', 5))

# part b
# all IP_protocols
print(top_n('IP_protocol', 20))

# part c
# top 5 destination ports
```

```
print("\nTop 5 Destination Ports: ")
print(top_n('udp_dst_port/tcp_dst_port/icmp_code', 5))

# part d
print("\n Total traffic in megabytes: ")
print(df['IP_packet_size'].sum() * 2048 / 1000000)

# part e
print("\n(E)")
# top 5 communication pairs
print("\n Top 5 communication pairs by IP address: ")
top_5_comm_pairs = top_n(['src_IP', 'dst_IP'], 5)
print(top_5_comm_pairs)

# all ethernet_types
print(top_n('ethernet_type', 10))



# Network graph for top 50 communication pairs

top_50_comm_pairs = top_n(['src_IP', 'dst_IP'], 50)

# Initialize a multi-directed graph
G = nx.MultiDiGraph()

# Add edges to the graph with weights based on the frequency of communication
for index, (src, dst, count) in top_50_comm_pairs.iterrows():
  G.add_edge(src, dst, weight=count)

# Draw the network graph
plt.figure(figsize=(12, 8))
pos = nx.spring_layout(G, k=1.5, iterations=16)
nx.draw_networkx_nodes(G, pos, node_size=700, node_color='skyblue')
nx.draw_networkx_edges(G, pos, width=[d['weight']/1000 for (u, v, d) in G.edges(data=True)],
edge_color='red', node_size=700)
nx.draw_networkx_labels(G, pos, font_size=10)
```

```
plt.title('Network Graph of top 50 IP Communication Pairs')
plt.axis('off')
plt.show()
```