INTERNET SERVICE PROVIDER (ISP)

CSE301 – COMPUTER NETWORKS

SUBMISSION: 01

GROUP - 04

13.08.2021

TEAM

NAME	ROLL NUMBER	CONTRIBUTION
Harini S	CB.EN.U4CSE19023	• Sales Analysis • Calculate Revenue • Add Attribute
Rithika Sri J	CB.EN.U4CSE19025	• Today's Feedback • Rating Analysis • Week's Performance
Nirmal K	CB.EN.U4CSE19038	• Cabling statistics • Bandwidth data • Server connectivity

DESCRIPTION:

Starlink is a satellite internet constellation operated by SapceX providing satellite internet access to most of the Earth. Our project focuses on creating a small scale Starlink to understand the possible networking aspects and challenges faced.

PERFORMANCE METRICS:

(1) Latency

Latency is a measure of delay. In a network, latency measures the time it takes for some data to get to its destination across the network. It is usually measured as a round trip delay - the time taken for information to get to its destination and back again. The round trip delay is an important measure because a computer that uses a TCP/IP network sends a limited amount of data to its destination and then waits for an acknowledgement to come back before sending any more. Thus, the round trip delay has a key impact on the performance of the network.

- **Latency = Propagation Time + Transmission Time + Queuing Time + Processing Delay** (Latency is usually measured in milliseconds (ms))
- Typical, approximate, values for latency that you might experience include:
  I. 800ms for satellite
  II. 120ms for 3G cellular data
  III. 60ms for 4G cellular data which is often used for 4G WAN and internet connections
  IV. 20ms for an mpls network such as BT IP Connect, when using Class of Service to prioritise traffic

V. 10ms for a modern Carrier Ethernet network

(2) Packet loss:

Examines how many data packets are dropped during data transmissions on your network.

The more data packets that are lost, the longer it takes for a data request to be fulfilled.

(3) Throughput:

- Measures your network's actual data transmission rate, which can vary wildly through different areas of your network. Indicates the percentage of data packets that are successfully being sent; a low throughput means there are a lot of failed or dropped packets that need to be sent again.
- **Throughput = TransferSize / TransferTime**
  TransferTime = RTT + 1/Bandwidth x TransferSize

- Round Trip Time (RTT) - time taken for a very small packet to travel across the network and return

(4) PROPAGATION TIME :

- Time required for a bit to travel from the source to the destination.
- **Propagation time = Distance / Propagation speed**
-

- **In wireless communication, $s=c$, i.e. the speed of light. In copper wire, the speed s generally ranges from .59c to .77c.**

NEED FOR NETWORKING IN AN ISP OFFICE:

File sharing: A network makes it easy for everyone to access the same file and prevents people from accidentally creating different versions.

Printer sharing: If you use a computer, chances are you also use a printer. With a network, several computers can share the same printer. Although you might need a more expensive printer to handle the added workload, it's still cheaper to use a network printer than to connect a separate printer to every computer in your office.

Communication and Collaboration: It's hard for people to work together if no one knows what anyone else is doing. A network allows employees to share files, view other people's work, and exchange ideas more efficiently. In a larger office, you can use email and instant messaging tools to communicate quickly and to store messages for future reference.

Organization: A variety of scheduling software is available that makes it possible to arrange meetings without constantly checking everyone's schedules. This software usually includes

other helpful features, such as shared address books and to-do lists.

Remote access: Having your own network allows greater mobility while maintaining the same level of productivity. With remote access in place, users are able to access the same files, data, and messages even when they're not in the office. This access can even be given to mobile handheld devices.

Data protection: You should know by now that it's vital to back up your computer data regularly. A network makes it easier to back up all of your company's data on an offsite server, a set of tapes, CDs, or other backup systems.

1. **Type of Network** : WAN [Wide Area Network]

2. **Client Configuration** :

   10th Gen Intel® Core™ i5-10600KF processor(6-Core, 12M Cache, 4.1GHz to 4.8GHz)
   8GB, 1x8GB, DDR4, 2666MHz
   512GB M.2 PCIe NVMe Solid State Drive
   500W Chassis

3. **Server Configuration** :

Database Server:

PowerEdge R340 Server
PowerEdge R340 MLK Motherboard
Intel® Xeon® E-2224 3.4GHz, 8M cache, 4C/4T, turbo (71W)
16GB UDIMM, 3200MT/s, ECC
18TB Hard Drive SATA 6Gbps 7.2K 512e 3.5in Hot-Plug
On-Board Broadcom 5720 Dual Port 1Gb LOM

Print Server:

5th Gen Intel® Core™ i5-5500K processor(4-Core, 6M Cache, 3.5GHz to 4.1GHz)
8GB, 2x4GB, DDR3, 1333MHz
4TB Seagate NAS grade Hard Disk Drive
300W Chassis

FTP Server:

5th Gen Intel® Core™ i5-5500K processor(4-Core, 6M Cache, 3.5GHz to 4.1GHz)
8GB, 2x4GB, DDR3, 1333MHz
18TB Seagate NAS grade Hard Disk Drive
300W Chassis

Mail Server:

7th Gen Intel® Core™ i7-7700K processor(4-Core, 10M Cache, 3.7GHz to 4.1GHz)
8GB, 2x4GB, DDR4, 2666MHz

1TB Seagate Barracuda Hard Disk Drive

400W Chassis

Proxy Server:

7th Gen Intel® Core™ i7-7700K processor(4-Core, 10M Cache, 4.1GHz to 4.4GHz)

8GB, 2x4GB, DDR4, 2400MHz

1TB Seagate Western Digital Black Hard Disk Drive

400W Chassis

Application Server:

HPE ProLiant DL325 Gen10 Plus v2 server with one AMD EPYC™ 7313P Processor,

32 GB DDR4 2666 MHz ECC memory,

HPE Smart Array P408i-a SR Gen10 Controller,

Eight small form factor drive bays - 20TB Seagate Enterprise Hard Disk Drives

500W power supply

CDN Server:

HPE ProLiant DL345 Gen10 Plus server with one AMD EPYC™ 7232P Processor

32 GB memory,

HPE Smart Array P408i-a SR Gen10 Controller,

Eight small form factor drive bays - 40TB Seagate Enterprise Hard Disk Drives

500W power supply

4. **Types of Servers** :

- Database server
- Print server
- FTP server
- Mail server
- Proxy server
- Application server
- CDN servers

5. **Network Medium** : Wireless / Coaxial cable / Fiber optical cable

6. **Topology** : Hybrid (mesh/star)

7.**Departments and number of users :**

- Customer care = 10 users
- Manager = 1 user
- Software Team = 20 users
- Network Team = 15 users
- Marketing team = 10 users

8. **Description for each department :**

- Customer care :

Handling feedback and complaints from customers. Responsible for establishing the relationship between the company and the public.

- Manager :

In charge of that particular branch. Responsible for planning, organizing, leading and controlling all the departments belonging to that branch.

- Software Team :

Web app management. Responsible for designing and maintaining web pages and applications related to the company.

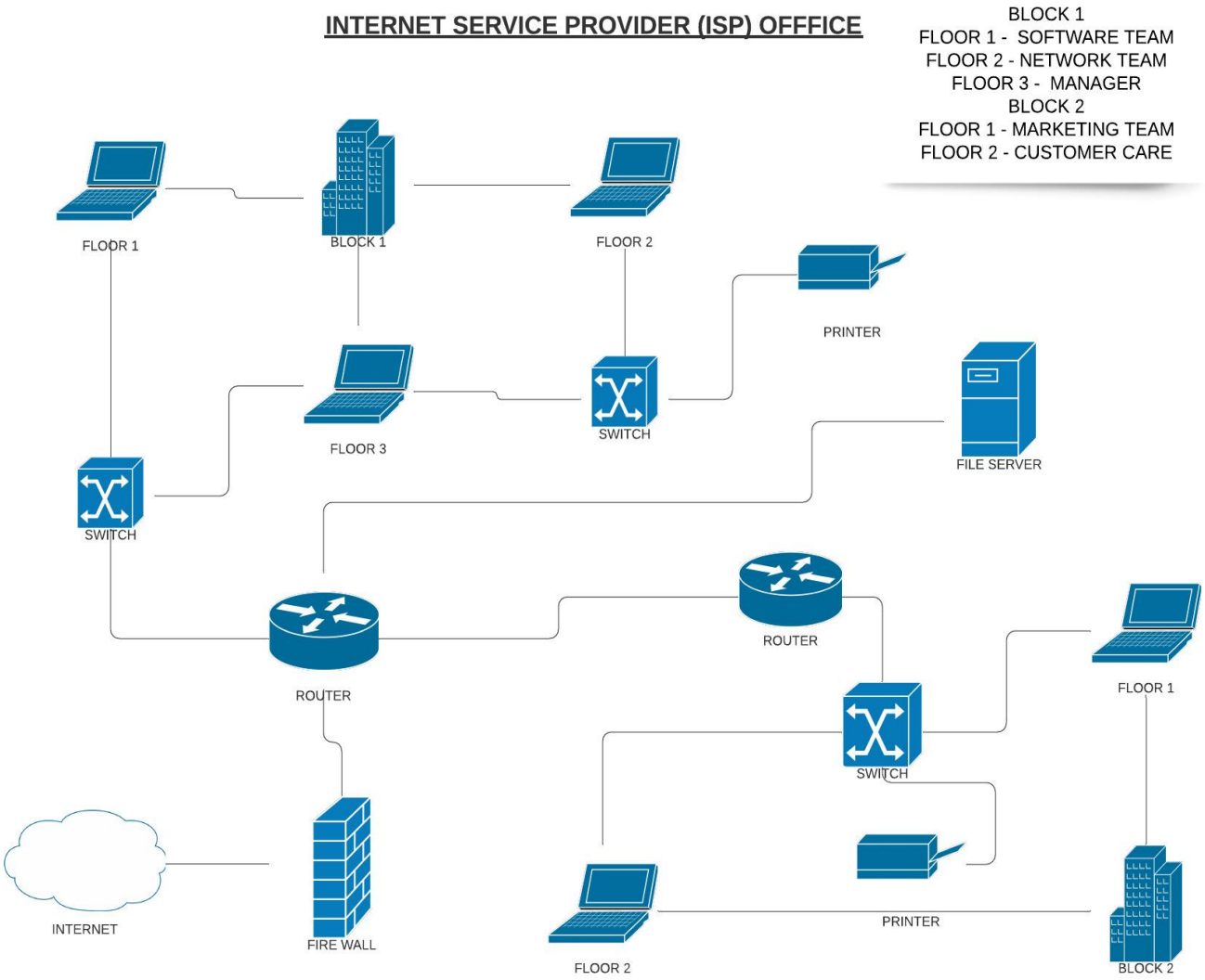
- Network Team :

Provide networking services to the other employees, customers and troubleshoot any problem that arises related to connections. Responsibility for setting up, developing and maintaining computer networks within an organisation.

- Marketing team :

Promoting the business. Responsible for creating and disseminating images, messages and ideas that best communicate the company value.

9. **Architecture diagram** :

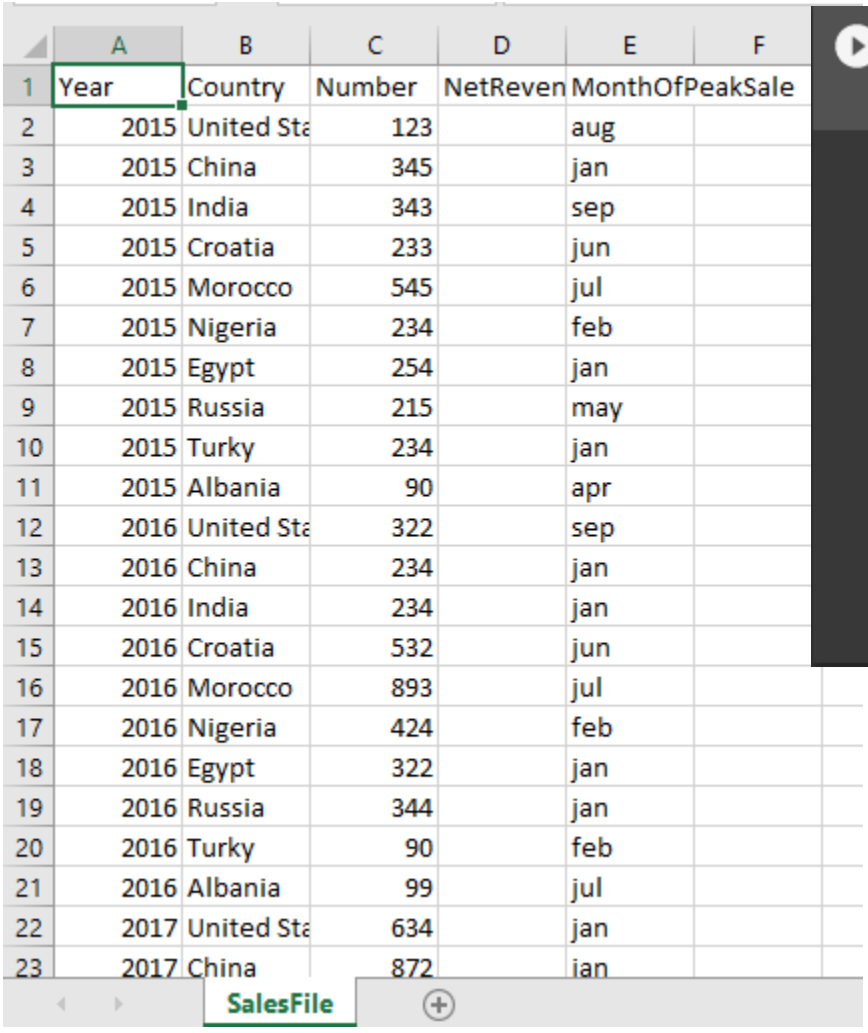


FILE HANDLING OPERATIONS USING SOCKET PROGRAMMING:

A. DESCRIPTION OF THE FILE:

SalesFile.csv

SalesFile.csv contains the sales report of the **internet connections sold** to different **countries** over the **years** and calculates the **total revenue** based on sales. It also has the **month of peak sale**. SalesFile.csv aims in the analysis of data to predict and increase the connections provided.



	A	B	C	D	E	F
1	Year	Country	Number	NetReven	MonthOfPeakSale	
2	2015	United Sta	123		aug	
3	2015	China	345		jan	
4	2015	India	343		sep	
5	2015	Croatia	233		jun	
6	2015	Morocco	545		jul	
7	2015	Nigeria	234		feb	
8	2015	Egypt	254		jan	
9	2015	Russia	215		may	
10	2015	Turky	234		jan	
11	2015	Albania	90		apr	
12	2016	United Sta	322		sep	
13	2016	China	234		jan	
14	2016	India	234		jan	
15	2016	Croatia	532		jun	
16	2016	Morocco	893		jul	
17	2016	Nigeria	424		feb	
18	2016	Egypt	322		jan	
19	2016	Russia	344		jan	
20	2016	Turky	90		feb	
21	2016	Albania	99		jul	
22	2017	United Sta	634		jan	
23	2017	China	872		jan	

SalesFile

```
file = pd.read_csv(FILE)
print(file.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70 entries, 0 to 69
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Year           70 non-null     int64
 1   Country        70 non-null     object
 2   Number         70 non-null     int64
 3   NetRevenue     0 non-null      float64
 4   MonthOfPeakSale 70 non-null    object
dtypes: float64(1), int64(2), object(2)
memory usage: 2.9+ KB
None
```

B. LIST OF **OPERATIONS** COMPLETED WITH THE FILE

1. **Sales Analysis Graph:**
   Minimum and maximum connections sold grouped by year
2. **Net Revenue Calculation:**
   A simple calculation of the total revenue (Number*$99)
3. **Add Attributes:**
   Addition of new features for future sale analysis

C. SERVER PROGRAM:

>> Combined

(Server.py)

```
'''
TOPIC: STARLINK
TEAM: {19023: "Harini S", 19025:"Rithika Sri J", 19038:"Nirmal K"}
SUBMISSION 1
'''
#SERVER PROGRAM

import socket
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import os
import _thread as thread
import string
import datetime
from numpy.lib.utils import info
import sys
import threading
import pickle
```

```
host = socket.gethostname() #dynamic IP
port1 = 5002
port2 = 5001
port3 = 5003

DISCONNECT_MESSAGE = "DISCONNECT"
FORMAT = "utf-8"
SIZE = 2048

FILE ="SalesFile.csv"
file = pd.read_csv(FILE)
df = pd.read_csv('customer_care.csv',index_col=0)
Items = pd.read_csv("./Bandwidth.csv")
Cabling = pd.read_csv("./Cabling.csv")

'''
----Harini S----
'''

def salesAnalysis(conn):
   try:
      yr = list(sorted(set(file["Year"])))
      a=list(file.groupby(['Year'])['Number'].max())
      b=list(file.groupby(['Year'])['Number'].min())
      x = np.arange(len(yr))
      width = 0.35

      fig, ax = plt.subplots()
      rects1 = ax.bar(x - width/2,a, width, label='Max')
      rects2 = ax.bar(x + width/2, b, width, label='Min')

      ax.set_ylabel('Client Count')
      ax.set_xlabel('Year')
      ax.set_xticks(x)
      ax.set_xticklabels(yr)
      ax.set_title('Sales Analysis')
      ax.legend()
      fig.tight_layout()
```

```
        fig.canvas.draw_idle()
        plt.show()
        op = plt.savefig("Sales.jpg")
        conn.send(bytes("[Successful] Sales Log Generated ",FORMAT))
    except:
        conn.send(bytes("[Fail] Error",FORMAT))

def revenue(conn):
    try:
        file['NetRevenue'].fillna((file["Number"] * 99), inplace=True)
        file.to_csv(FILE, index=False)
        conn.send(bytes(FILE,FORMAT))
    except:
        conn.send(bytes("[Fail] Error",FORMAT))


def addAttr(dataSet,conn):
    try:
        fname = os.path.basename(dataSet)
        fileNew = pd.read_csv(fname)
        tempFile=pd.concat([file,fileNew], axis = 1)
        tempFile.to_csv(FILE, index=False)
        conn.send(bytes(FILE,FORMAT))
    except:
        conn.send(bytes("[Fail] Error",FORMAT))

def mclient(conn, addr, host):
    print("[CONNECTION] Server Connected ", addr[0])
    while True:
        inp = (conn.recv(SIZE).decode(FORMAT)).split(" ")
        if inp[0] =="0":
            break
        if inp[0] =="1":
            salesAnalysis(conn)
        elif inp[0] =="2":
            revenue(conn)
        elif inp[0] == "3":
            addAttr(inp[1],conn)
```

```
    conn.close()

'''
----Rithika Sri J----
'''
def feedback_today(conn_socket, date):
    new_df = df.loc[df['DATE'] == date]
    new_df.to_csv('today_feedback.csv',index=False)
    conn_socket.send(bytes(new_df.to_string(), 'utf-8'))
    try:
        pass
    except:
        conn_socket.send(b"Operation failed!")

def rating_analysis(conn_socket):
    y=list(df["RATING"].value_counts())
    print(df['RATING'],df["RATING"].value_counts())
    x=[5,4,3,2,1]
    plt.bar(x,y,color='r')
    plt.title("CUSTOMER RATING")
    plt.xlabel("Stars")
    plt.ylabel("Number of customers")
    plt.savefig('rating.png', dpi=300, bbox_inches='tight')
    conn_socket.send(bytes('rating.png', 'utf-8'))
    try:
        pass
    except:
        conn_socket.send(b"Operation failed!")

def week_performance(conn_socket, d1,d2):
    week=[]
    fd=open("Week_performance.txt",'w')
    fd.write("Last week's DATE (from): "+ str(d1) + "\nToday's DATE (to): "+str(d2))
    d1 = datetime.datetime.strptime(d1, '%d-%m-%Y')
    d2 = datetime.datetime.strptime(d2, '%d-%m-%Y')
    step = datetime.timedelta(days=1)
    while d2 <= d1:
        week.append(d2.strftime("%d-%m-%Y"))
```

```
        d2 += step
    new_df= df.loc[df['DATE'].isin(week)]
    perc= ((new_df['RATING'].sum())/(5*len(new_df)))*100
    info= "\nMEAN : "+ str(new_df['RATING'].mean(axis=0)) + "\nMAX : "+
str(new_df['RATING'].max()) + "\nMIN : "+ str(new_df['RATING'].min()) + "\n\t***WEEK'S
PERFORMANCE RATE*** : "+str(perc)+"%"
    fd.write(info)
    fd.close()
    conn_socket.send(bytes(info, 'utf-8'))
    try:
        pass
    except:
        conn_socket.send(b"Operation failed!")

def on_new_client(clientsocket, addr, host):
    while True:
        msg = clientsocket.recv(1024).decode()
        args = msg.split(SEPARATOR)

        if(args[0] == "-1"):
            break
        elif(args[0] == "0"):
            feedback_today(clientsocket, args[1])
        elif(args[0] == "1"):
            rating_analysis(clientsocket)
        elif(args[0] == "2"):
            week_performance(clientsocket, args[1], args[2])
    clientsocket.close()

'''
----Nirmal K----
'''
def Log(inp):
    now = datetime.now()
    current_time = now.strftime("%H:%M:%S")
    print("[", current_time, "] Operation ", inp, " has been initiated")
```

```
def Log_finish(inp, status):
    now = datetime.now()
    current_time = now.strftime("%H:%M:%S")

    if status == True:
        print("[", current_time, "] Operation ", inp, " has been successfully executed")

    else:
        print("[", current_time, "] Operation ", inp, " has failed")


'''
Multi-Socket
'''
print("[STARTING] Server Starting")
server1 = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server2 = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server3 = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

server1.bind((host,port1))
server2.bind((host,port2))
server3.bind((host,port3))

server1.listen(5)    #backlog = 5
server2.listen()
server3.listen()
print("[LISTENING] Server Listening")


while True:
    try:
        conn, addr = server1.accept() # addr = (host,port)
        thread.start_new_thread(mclient, (conn, addr, host))
    except:
        try:
            c, addr = server2.accept()
            thread.start_new_thread(on_new_client, (c, addr, host))
        except:
```

```
client, addr = server3.accept()
Items = pd.read_csv("./Bandwidth.csv")
print(f'New connection established with {addr}')

inp = (client.recv(SIZE).decode('utf-8'))
if(inp=="Enter"):
    Log(inp)
    data = client.recv(SIZE)
    recd = pickle.loads(data)
    print("Values recieved: ")
    print(recd)
    Items = Items.append(recd,ignore_index=True)
    Log_finish(inp,status=True)

elif(inp == "View"):
    Log(inp)
    send_file = pickle.dumps(Items)
    stat = client.send(send_file)
    if(stat==0):
        Log_finish(inp,status=False)
    else:
        Log_finish(inp,status=True)

elif(inp == "Overload"):
    Log(inp)
    df = Items.loc[(Items["Allocation"]<Items["Usage"])]
    send_file = pickle.dumps(Items)
    stat = client.send(send_file)
    if(stat==0):
        Log_finish(inp,status=False)
    else:
        Log_finish(inp,status=True)

elif(inp == "Insert"):
    Log(inp)
    data = client.recv(SIZE)
    recd = pickle.loads(data)
    print("Values recieved: ")
```

```
            print(recd)
            Cabling = Cabling.append(recd,ignore_index=True)
            print(Cabling)
            Log_finish(inp,status=True)

        elif(inp == "Type"):
            Log(inp)
            cable = client.recv(SIZE)
            df = Cabling.loc[(Items["type"] == cable)]
            send_file = pickle.dumps(Items)
            stat = client.send(send_file)
            if(stat==0):
                Log_finish(inp,status=False)
            else:
                Log_finish(inp,status=True)

        elif(inp == "list_by_region"):
            Log(inp)
            region = client.recv(SIZE)
            df = Cabling.sort_values(by = 'region')
            send_file = pickle.dumps(Items)
            stat = client.send(send_file)
            if(stat==0):
                Log_finish(inp,status=False)
            else:
                Log_finish(inp,status=True)



server1.close()
server2.close()
server3.close()
```

D. CLIENT PROGRAM:

```
import socket
import pandas as pd
import os
FORMAT = "utf-8"
```

```
SIZE = 2048

host = socket.gethostname() #dynamic IP
port = 5002
client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.connect((host,port))

lst = {0:"Disconnect",1:"Generate Sales Graph", 2:"Calculate Revenue", 3:"Add Attribute --
Dataset"}

for i, j in lst.items():
   print(i,": ",j)
print()
inp = input()

if inp == "3":
   dfile = (input("Dataset: \n"))
   inp = inp+" "+dfile
   client.send(bytes(inp,FORMAT))
   inp = "3"
else :
   client.send(bytes(inp,FORMAT))


if inp == "2" or inp == "3":
   f =  client.recv(SIZE).decode(FORMAT)
   fname = os.path.basename(f)
   file = pd.read_csv(fname)
   print("FILE INFO:\n")
   print(file.info())
   print("\nFILE CONTENT\n")
   print(file)
   print("[Successful] Updated ")
else:
   print (client.recv(SIZE).decode(FORMAT))
client.close()
```

OUTPUT:

>>Original File (SalesFile.csv)

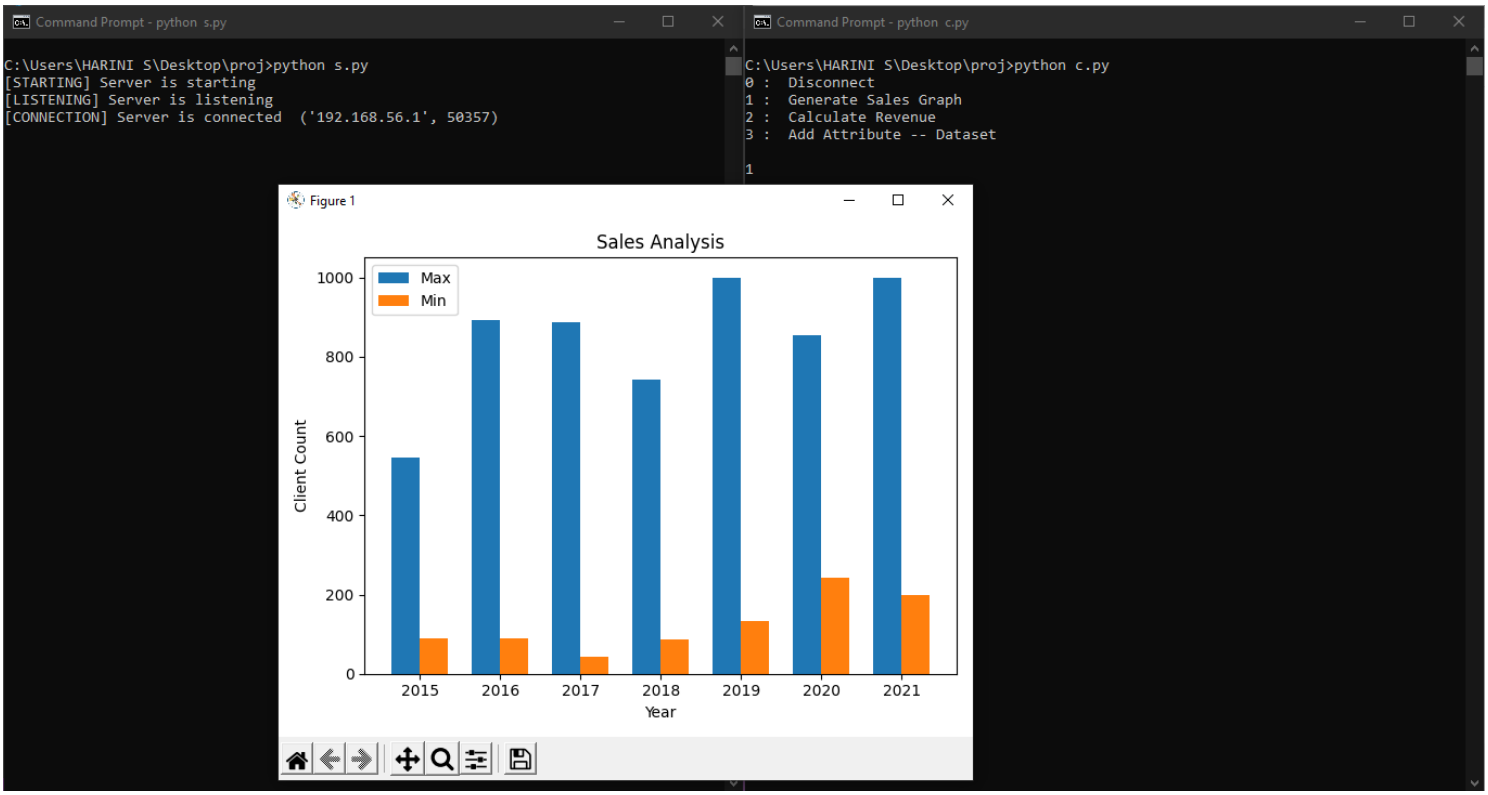
	A	B	C	D	E	F
1	Year	Country	Number	NetReven	MonthOfPeakSale	
2	2015	United Sta	123		aug	
3	2015	China	345		jan	
4	2015	India	343		sep	
5	2015	Croatia	233		jun	
6	2015	Morocco	545		jul	
7	2015	Nigeria	234		feb	
8	2015	Egypt	254		jan	
9	2015	Russia	215		may	
10	2015	Turky	234		jan	
11	2015	Albania	90		apr	
12	2016	United Sta	322		sep	
13	2016	China	234		jan	
14	2016	India	234		jan	
15	2016	Croatia	532		jun	
16	2016	Morocco	893		jul	
17	2016	Nigeria	424		feb	
18	2016	Egypt	322		jan	
19	2016	Russia	344		jan	
20	2016	Turky	90		feb	
21	2016	Albania	99		jul	
22	2017	United Sta	634		jan	
23	2017	China	872		jan	

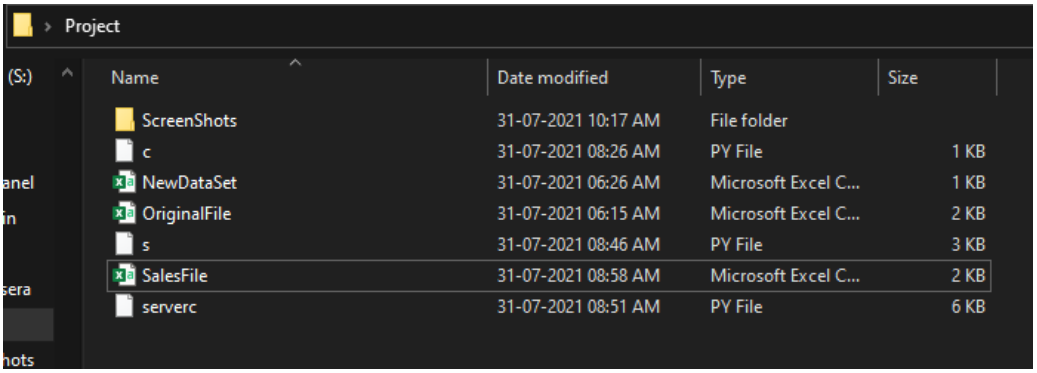
SalesFile

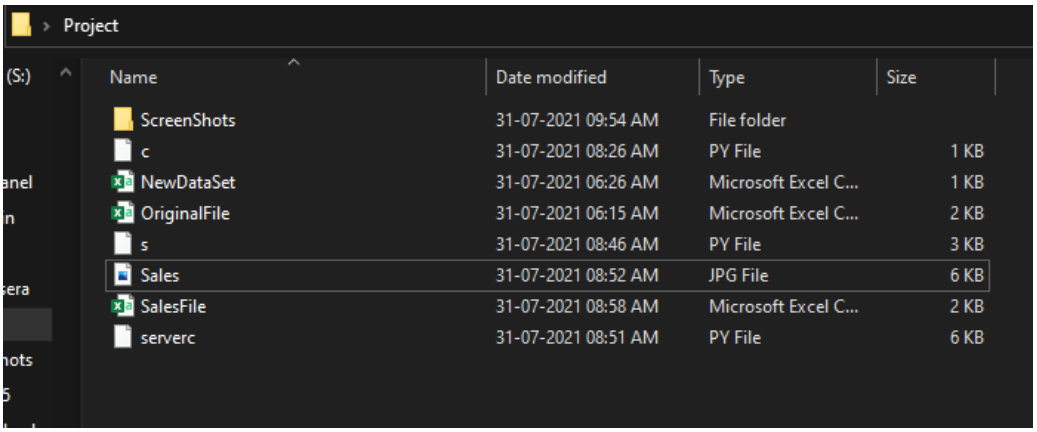
```
file = pd.read_csv(FILE)
print(file.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70 entries, 0 to 69
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Year            70 non-null     int64
 1   Country         70 non-null     object
 2   Number          70 non-null     int64
 3   NetRevenue      0 non-null      float64
 4   MonthOfPeakSale 70 non-null     object
dtypes: float64(1), int64(2), object(2)
memory usage: 2.9+ KB
None
```

>>Function 1: (Sales Analysis)







>>Function 2: (Calculate Net Revenue)

```
Command Prompt - python  s.py

C:\Users\HARINI S\Desktop\proj>python s.py
[STARTING] Server is starting
[LISTENING] Server is listening
[CONNECTION] Server is connected  ('192.168.56.1', 50357)
[CONNECTION] Server is connected  ('192.168.56.1', 50365)
```

	A	B	C	D	E	F
1	Year	Country	Number	NetReven	MonthOfPeakSale	
2	2015	United Sta	123	12177	aug	
3	2015	China	345	34155	jan	
4	2015	India	343	33957	sep	
5	2015	Croatia	233	23067	jun	
6	2015	Morocco	545	53955	jul	
7	2015	Nigeria	234	23166	feb	
8	2015	Egypt	254	25146	jan	
9	2015	Russia	215	21285	may	
10	2015	Turky	234	23166	jan	
11	2015	Albania	90	8910	apr	
12	2016	United Sta	322	31878	sep	
13	2016	China	234	23166	jan	
14	2016	India	234	23166	jan	
15	2016	Croatia	532	52668	jun	
16	2016	Morocco	893	88407	jul	
17	2016	Nigeria	424	41976	feb	
18	2016	Egypt	322	31878	jan	
19	2016	Russia	344	34056	jan	
20	2016	Turky	90	8910	feb	
21	2016	Albania	99	9801	jul	
22	2017	United Sta	634	62766	jan	
23	2017	China	872	86328	jan	

SalesFile

```
Command Prompt

C:\Users\HARINI S\Desktop\proj>python c.py
0 :  Disconnect
1 :  Generate Sales Graph
2 :  Calculate Revenue
3 :  Add Attribute -- Dataset

2
FILE INFO:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70 entries, 0 to 69
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Year            70 non-null     int64
 1   Country         70 non-null     object
 2   Number          70 non-null     int64
 3   NetRevenue      70 non-null     float64
 4   MonthOfPeakSale  70 non-null     object
dtypes: float64(1), int64(2), object(2)
memory usage: 2.9+ KB
None

FILE CONTENT

    Year        Country  Number  NetRevenue MonthOfPeakSale
0   2015  United States     123     12177.0             aug
1   2015          China     345     34155.0             jan
2   2015          India     343     33957.0             sep
3   2015        Croatia     233     23067.0             jun
4   2015        Morocco     545     53955.0             jul
..   ...            ...     ...         ...             ...
65  2021        Nigeria     500     49500.0             sep
66  2021          Egypt     200     19800.0             aug
67  2021         Russia     200     19800.0             jul
68  2021          Turky     260     25740.0             jun
69  2021        Albania     300     29700.0             dec

[70 rows x 5 columns]
[Successful] Updated

C:\Users\HARINI S\Desktop\proj>
```

>>Function 3: (Add Attributes) (NewDataSet.csv)

```
Command Prompt - python  s.py

C:\Users\HARINI S\Desktop\proj>python s.py
[STARTING] Server is starting
[LISTENING] Server is listening
[CONNECTION] Server is connected  ('192.168.56.1', 50357)
[CONNECTION] Server is connected  ('192.168.56.1', 50365)
[CONNECTION] Server is connected  ('192.168.56.1', 61050)
```

```
Command Prompt

1 :  Generate Sales Graph
2 :  Calculate Revenue
3 :  Add Attribute -- Dataset

3
Dataset:
NewDataSet.csv
FILE INFO:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70 entries, 0 to 69
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Year             70 non-null     int64
 1   Country          70 non-null     object
 2   Number           70 non-null     int64
 3   NetRevenue       70 non-null     float64
 4   MonthOfPeakSale  70 non-null     object
 5   SatCoverage      9 non-null      float64
dtypes: float64(2), int64(2), object(2)
memory usage: 3.4+ KB
None

FILE CONTENT

    Year        Country  Number  NetRevenue MonthOfPeakSale  SatCoverage
0   2015  United States     123     12177.0             aug        100.0
1   2015          China     345     34155.0             jan         50.0
2   2015          India     343     33957.0             sep         50.0
3   2015        Croatia     233     23067.0             jun         40.0
4   2015        Morocco     545     53955.0             jul        100.0
..   ...            ...     ...         ...             ...          ...
65  2021        Nigeria     500     49500.0             sep          NaN
66  2021          Egypt     200     19800.0             aug          NaN
67  2021         Russia     200     19800.0             jul          NaN
68  2021          Turky     260     25740.0             jun          NaN
69  2021        Albania     300     29700.0             dec          NaN

[70 rows x 6 columns]
[Successful] Updated

C:\Users\HARINI S\Desktop\proj>
```

	A
1	SatCoverage
2	100
3	50
4	50
5	40
6	100
7	70
8	80
9	90
10	80

NewDataSet

A1 : Year

	A	B	C	D	E	F
1	Year	Country	Number	NetReven	MonthOfP	SatCoverage
2	2015	United Sta	123	12177	aug	100
3	2015	China	345	34155	jan	50
4	2015	India	343	33957	sep	50
5	2015	Croatia	233	23067	jun	40
6	2015	Morocco	545	53955	jul	100
7	2015	Nigeria	234	23166	feb	70
8	2015	Egypt	254	25146	jan	80
9	2015	Russia	215	21285	may	90
10	2015	Turky	234	23166	jan	80
11	2015	Albania	90	8910	apr	
12	2016	United Sta	322	31878	sep	
13	2016	China	234	23166	jan	
14	2016	India	234	23166	jan	
15	2016	Croatia	532	52668	jun	
16	2016	Morocco	893	88407	jul	
17	2016	Nigeria	424	41976	feb	
18	2016	Egypt	322	31878	jan	
19	2016	Russia	344	34056	jan	
20	2016	Turky	90	8910	feb	
21	2016	Albania	99	9801	jul	
22	2017	United Sta	634	62766	jan	
23	2017	China	872	86328	jan	

SalesFile

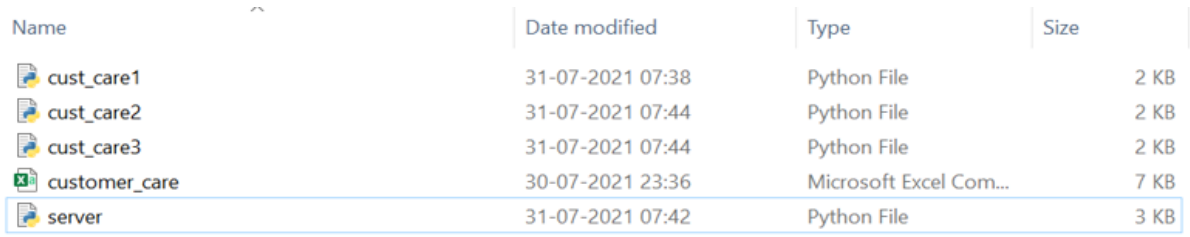
RITHIKA SRI J [CB.EN.U4CSE19025]

```
import socket
from datetime import date
from datetime import timedelta
from PIL import Image
SEPARATOR = ","

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
host = socket.gethostname()
port = 5001

s.connect((host, port))
lst = [-1, 0, 1, 2, 3]
while True:
   number = int(input("0: View Today's Feedback \n1: Rating Analysis \n2: Performance
rate of the week \nEnter option (-1 to break): "))

   if number in lst:
      send_str = str(number)
      if(number == 0):
         date=date.today()
         d1 = date.strftime("%d-%m-%Y")
         print("Today's DATE: "+ str(d1))
         send_str = send_str+SEPARATOR+d1
         s.send(bytes(send_str, 'utf-8'))
         print(s.recv(2048).decode())

      if(number == 1):
         s.send(bytes(send_str, 'utf-8'))
         print(s.recv(2048).decode())
         print("File received...")
         rating=Image.open('rating.png')
         print("File opened..")
         rating.show()

      if(number == 2):
         d1 = date.today()
         d2 = d1 - timedelta(days=7)
```

```
            d1 = d1.strftime("%d-%m-%Y")
            d2 = d2.strftime("%d-%m-%Y")
            print("Today's DATE: "+ str(d1))
            print("Last week's DATE: "+str(d2))
            send_str = send_str+SEPARATOR+d1+SEPARATOR+d2
            s.send(bytes(send_str, 'utf-8'))
            print(s.recv(2048).decode())

        if(number == -1):
            break
    else:
        print("Invalid number!")
    print("\n")
s.close()
```

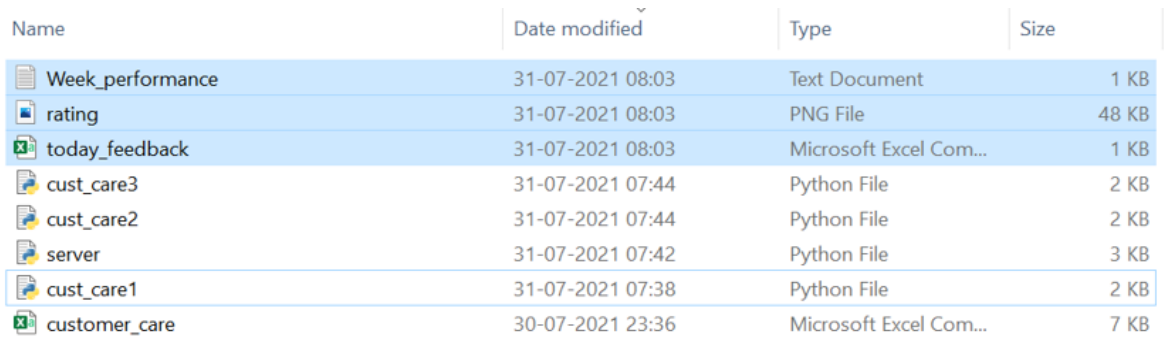
EXECUTION:

>> Before execution:



Name	Date modified	Type	Size
cust_care1	31-07-2021 07:38	Python File	2 KB
cust_care2	31-07-2021 07:44	Python File	2 KB
cust_care3	31-07-2021 07:44	Python File	2 KB
customer_care	30-07-2021 23:36	Microsoft Excel Com...	7 KB
server	31-07-2021 07:42	Python File	3 KB

>> After execution:



Name	Date modified	Type	Size
Week_performance	31-07-2021 08:03	Text Document	1 KB
rating	31-07-2021 08:03	PNG File	48 KB
today_feedback	31-07-2021 08:03	Microsoft Excel Com...	1 KB
cust_care3	31-07-2021 07:44	Python File	2 KB
cust_care2	31-07-2021 07:44	Python File	2 KB
server	31-07-2021 07:42	Python File	3 KB
cust_care1	31-07-2021 07:38	Python File	2 KB
customer_care	30-07-2021 23:36	Microsoft Excel Com...	7 KB

>> Customer_care.csv

	A	B	C	D	E	F	G
1	ID	DATE	NAME	PHONE	CITY	FEEDBACK/ISSUE	RATING
2	1	03-08-2021	Lia	9057919869	Toulouse	fbAAo	1
3	2	28-07-2021	Hamil	9394663171	CuiabÃ¡	rCEZ	2
4	3	02-08-2021	Parsaye	9731024503	Columbus	HRJLY	3
5	4	31-07-2021	Ruvolo	9632599148	Podgorica	IIJmI	4
6	5	03-08-2021	Kenwood	9944337572	Nouakchott	c ls	5
7	6	28-07-2021	Saunderson	9448116229	Charlotte Ama	fCS	5
8	7	02-08-2021	Podvin	9501829311	Vilnius	KYUAy	5
9	8	01-08-2021	Monk	9270242438	Chennai	OfOpq	3
10	9	31-07-2021	Regan	9038268745	Podgorica	Q zia	4
11	10	01-08-2021	Gert	9532127781	Dallas	eaBAj	1
12	11	28-07-2021	Nedrud	9668259783	DÃ¼sseldorf	uPHn	3
13	12	03-08-2021	Lea	9609065265	Jacksonville	xxxi	1
14	13	31-07-2021	Brunell	9310573035	Bratsk	PSVmI	5
15	14	03-08-2021	Rosemary	9126959382	Iqaluit	vlldj	5
16	15	01-08-2021	Clarissa	9210370855	MedellÃ-n	HpRs	5
17	16	30-07-2021	Olin	9143631994	Ä°zmir	aRVBD	1
18	17	02-08-2021	Rocray	9427748358	Saskatoon	lqaq	4
19	18	02-08-2021	Hepsibah	9087800813	Dushanbe	uKBS	4
20	19	03-08-2021	Bigner	9717210663	Philadelphia	esKx	4
21	20	02-08-2021	Fancie	9848658262	Lobamba	TTn G	1
22	21	03-08-2021	Marcellus	9716380115	Boa Vista	mmeT	5

```
ame.DataFrame'>
ies, 1.0 to nan
columns):
on-Null Count  Dtype
-------------  -----
00 non-null    object
00 non-null    object
00 non-null    float64
00 non-null    object
00 non-null    object
00 non-null    float64
ject(4)
memory usage: 11.7+ KB
```

Execution of a client:

>> FUNCTION 1 : 0 - To view today's Feedback/Issue

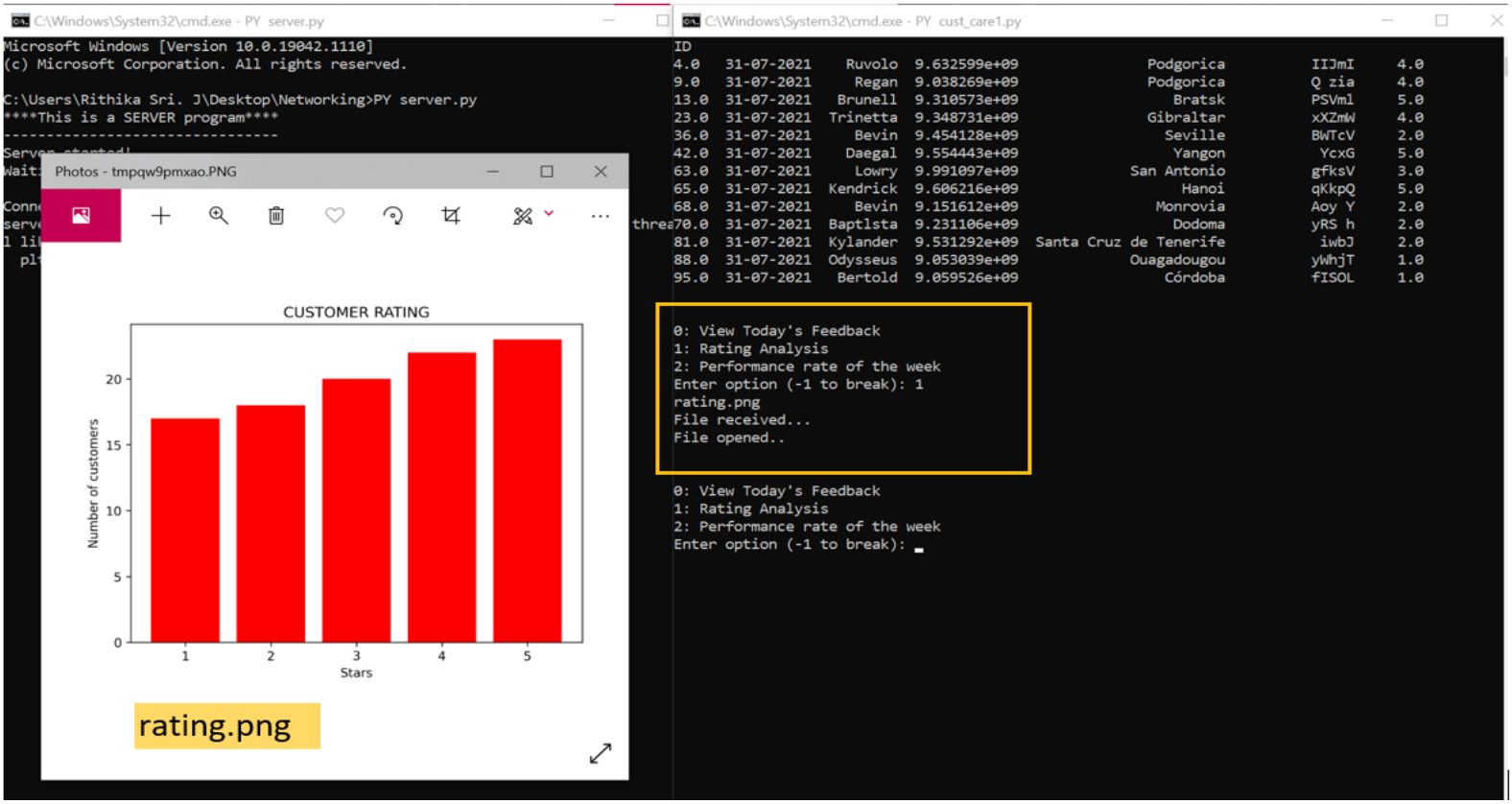
```
C:\Windows\System32\cmd.exe - PY server.py
Microsoft Windows [Version 10.0.19042.1110]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Rithika Sri. J\Desktop\Networking>PY server.py
****This is a SERVER program****
--------------------------------
Server started!
Waiting for customer care...

Connected to a customer care : 192.168.1.100
```

```
C:\Windows\System32\cmd.exe - PY cust_care1.py
Microsoft Windows [Version 10.0.19042.1110]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Rithika Sri. J\Desktop\Networking>PY cust_care1.py
0: View Today's Feedback
1: Rating Analysis
2: Performance rate of the week
Enter option (-1 to break): 0
Today's DATE: 31-07-2021
            DATE      NAME         PHONE                  CITY FEEDBACK/ISSUE  RATING
ID
4.0   31-07-2021     Ruvolo  9.632599e+09             Podgorica         IIJmI     4.0
9.0   31-07-2021      Regan  9.038269e+09             Podgorica         Q zia     4.0
13.0  31-07-2021    Brunell  9.310573e+09               Bratsk         PSVmI     5.0
0     31-07-2021   Trinetta  9.348731e+09            Gibraltar         xXZmW     4.0
0     31-07-2021      Bevin  9.454128e+09              Seville         BWTcV     2.0
0     31-07-2021     Daegal  9.554443e+09               Yangon          YcxG     5.0
0     31-07-2021      Lowry  9.991097e+09            San Antonio         gfksV     3.0
0     31-07-2021   Kendrick  9.606216e+09               Hanoi         qKkpQ     5.0
0     31-07-2021      Bevin  9.151612e+09             Monrovia         Aoy Y     2.0
0     31-07-2021    Baptlsta  9.231106e+09              Dodoma         yRS h     2.0
0     31-07-2021    Kylander  9.531292e+09  Santa Cruz de Tenerife          iwbJ     2.0
0     31-07-2021    Odysseus  9.053039e+09            Ouagadougou         yWhjT     1.0
0     31-07-2021     Bertold  9.059526e+09               Córdoba         fISOL     1.0

View Today's Feedback
Rating Analysis
Performance rate of the week
er option (-1 to break): _
```

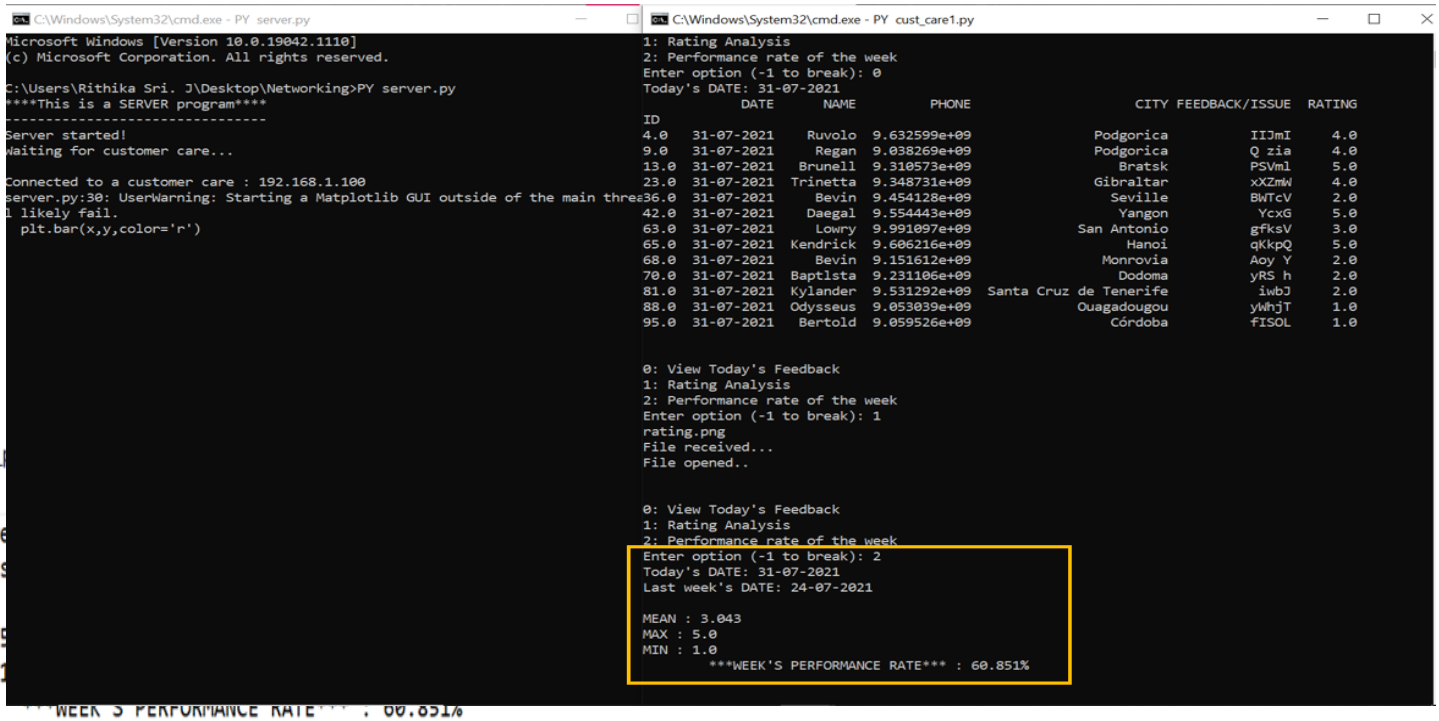
	A	B	C	D	E	F
1	DATE	NAME	PHONE	CITY	FEEDBACK/ISSUE	RATING
2	31-07-2021	Ruvolo	9632599148	Podgorica	IIJmI	4
3	31-07-2021	Regan	9038268745	Podgorica	Q zia	4
4	31-07-2021	Brunell	9310573035	Bratsk	PSVmI	5
5	31-07-2021	Trinetta	9348731205	Gibraltar	xXZmW	4
6	31-07-2021	Bevin	9454128203	Seville	BWTcV	2
7	31-07-2021	Daegal	9554442571	Yangon	YcxG	5
8	31-07-2021	Lowry	9991096904	San Antonio	gfksV	3
9	31-07-2021	Kendrick	9606216465	Hanoi	qKkpQ	5
10	31-07-2021	Bevin	9151612229	Monrovia	Aoy Y	2
11	31-07-2021	Baptlsta	9231105817	Dodoma	yRS h	2
12	31-07-2021	Kylander	9531291920	Santa Cruz de Tenerife	iwbJ	2
13	31-07-2021	Odysseus	9053038782	Ouagadougou	yWhjT	1
14	31-07-2021	Bertold	9059525905	CÃ³rdoba	fISOL	1

Today_feedback.csv

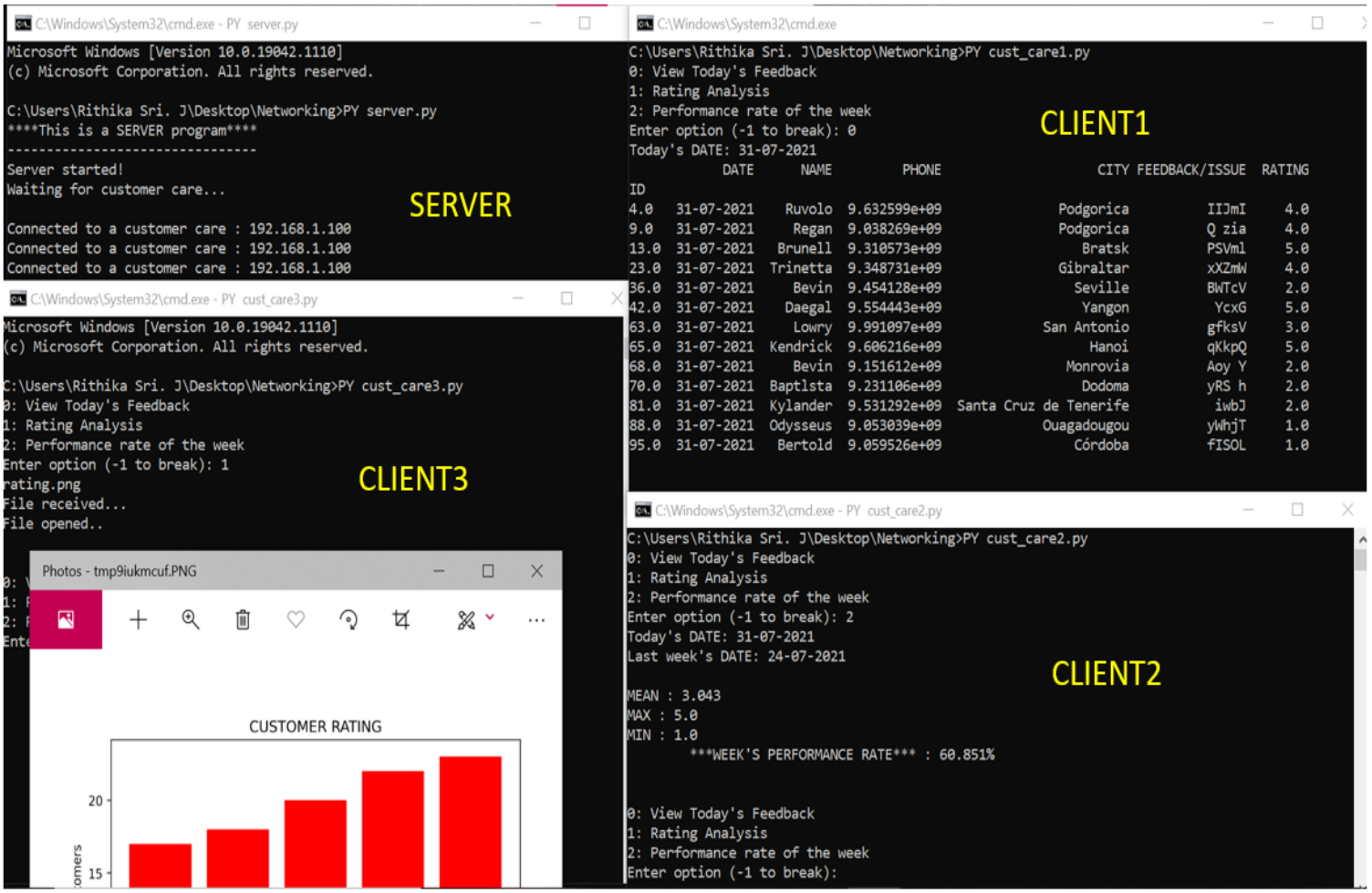
>> FUNCTION 2: 1 - Rating Analysis



>> FUNCTION 3: 2 - Week's Performance



Working with multiple clients:



NIRMAL K [CB.EN.U4CSE19038]

```
 import socket
import numpy as np
import pandas as pd
import pickle

SERVER = socket.gethostname()
PORT = 5003
ADDR = (SERVER,PORT)
CHUNK_SIZE = 2048

FORMAT = 'utf-8'
DISCONNECT_MESSAGE = "DISCONNECT"

client = socket.socket(socket.AF_INET,socket.SOCK_STREAM)

try:
    client.connect(ADDR)
except socket.error as error:
    print(
        "Server seems to be down. Sorry for the inconvinience, please try sometime later"
        + "\n" + str(error)
    )


cont = "Y"

def bandwidth():
    print("Bandwidth")
    things_to_do = ["Enter data (E)", "View by region [V]", "Check overload (O)"]

    for x in things_to_do:
        print(x)

    choice = input("Enter your choice: ")
```

```
    if choice == "E":
        func = "Enter".encode('utf-8')
        client.send(func)
        print("Enter values to be inserted: ")
        date = input("Date(dd-mm-yyyy): ")
        region = input("Region: ")
        alloc = int(input("Allocation: "))
        exp_usage = int(input("Expected usage: "))
        usage = int(input("Usage"))

        insert_data = {'Usage':usage, 'allocation':alloc, 'expusage':exp_usage, 'region':region,
'year':date}

        msg = pickle.dumps(insert_data)
        msg = bytes(msg)
        status = client.send(msg)
        print("Status of sending: ",status)

    if choice == "V":
        func = "View".encode(FORMAT)
        client.send(func)
        msg = client.recv(CHUNK_SIZE)
        recd = pickle.loads(msg)
        print(recd)

    if choice == "O":
        func = "Overload".encode(FORMAT)
        client.send(func)
        msg = client.recv(CHUNK_SIZE)
        recd = pickle.loads(msg)
        print(recd)

def cabling():
    print("Cabling statistics")
    print("Choose operation: ")
    options = ["Inspection (I)", "Type (T)", "List by region (R)"]
```

```
    for x in options:
        print(x)

    choice = input()

    if choice == "I":
        func = "Insert".encode('utf-8')
        client.send(func)
        print("Enter values to be inserted: ")
        region = input("Region: ")
        length = input("Length of cabling: ")
        type = input("Type of cable: ")
        cost = input("Cost of installation: ")
        dateofinstall = input("Date of installation: ")
        issuecount = input("Number of issues reported: ")
        checkdate = input("Last date of inspection: ")
        nextcheck = input("Next check date: ")

        insert_data = {'region':region, 'length':length, 'type':type, 'cost':cost ,
'dateofinstall':dateofinstall, 'issuecount':issuecount, 'checkdate':checkdate,
'nextcheck':nextcheck}

        msg = pickle.dumps(insert_data)
        msg = bytes(msg)
        status = client.send(msg)
        print("Status of sending: ",status)

    elif choice == "T":
        print("Choose cable type: ")
        print("1. CAT4\n2. CAT5\n3. CAT6\n4. CAT6e\n5. Fiber optic\n6. Satellite")
        cable = input("Enter number: ")
        if(cable==1):
            send = "cat4".encode('utf-8')
            client.send(send)
            msg = client.recv(CHUNK_SIZE)
            recd = pickle.loads(msg)
            print(recd)
```

```
elif(cable==2):
    send = "cat5".encode('utf-8')
    client.send(send)
    msg = client.recv(CHUNK_SIZE)
    recd = pickle.loads(msg)
    print(recd)

elif(cable==3):
    send = "cat6".encode('utf-8')
    client.send(send)
    msg = client.recv(CHUNK_SIZE)
    recd = pickle.loads(msg)
    print(recd)

elif(cable==4):
    send = "cat6e".encode('utf-8')
    client.send(send)
    msg = client.recv(CHUNK_SIZE)
    recd = pickle.loads(msg)
    print(recd)

elif(cable==5):
    send = "fiber".encode('utf-8')
    client.send(send)
    msg = client.recv(CHUNK_SIZE)
    recd = pickle.loads(msg)
    print(recd)

elif(cable==6):
    send = "satellite".encode('utf-8')
    client.send(send)
    msg = client.recv(CHUNK_SIZE)
    recd = pickle.loads(msg)
    print(recd)

else:
    print("Please choose one of the available options")
    return
```

```
    elif choice == "R":
        send = "list_by_region".encode('utf-8')
        client.send(send)
        msg = client.recv(CHUNK_SIZE)
        recd = pickle.loads(msg)
        print(recd)


    else:
        print("Try again, invalid option")

    print("Do you want to continue? (Y/N)", end=" ")
    cont = input()

def server():
    p = 80
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    host_ip = socket.gethostbyname('www.google.com')
    print("Google:",host_ip)
    status = s.connect(host_ip,p)
    print(status, ": Status of connection")


while True and cont=="Y":
    print("Networking command center")
    print("Choose operation: ")
    options = ["Bandwidth (B) ","Cabling statistics (C) ","Check server status (S) "]

    for x in options:
        print(x)

    choice = input()

    if choice == "B":
        bandwidth()
```

```
elif choice == 'C':
    cabling()


elif choice == "S":
    server()
    break

else:
    print("Try again, invalid option")

print("Do you want to continue? (Y/N)", end=" ")
cont = input()
```

EXECUTION:

```
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWi
ndows

PS E:\Projects\Semester5\Networking\Project> & C:/Users/nirma/AppData/Local/Microsof
t/WindowsApps/python3.9.exe e:/Projects/Semester5/Networking/Project/server.py
Server starting up.
Server address:  ('DESKTOP-H35TRG8', 5001)
New connection established with ('192.168.56.1', 7416)
```

```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWi
ndows

PS E:\Projects\Semester5\Networking> & C:/Users/nirma/AppData/Local/Microsoft/Window
sApps/python3.9.exe e:/Projects/Semester5/Networking/Project/client_networking.py
Networking command center
Choose operation:
Bandwidth (B)
Cabling statistics (C)
Check server status (S)
```

Accounts

```
Cabling statistics (C)
Check server status (S)
B
Bandwidth
Enter data (E)
View by region [V]
Check overload (O)
Enter your choice: E
Enter values to be inserted:
Date(dd-mm-yyyy): 12-08-2021
Region: Southwest
Allocation: 12000
Expected usage: 11900
Usage11915
Status of sending:  98
Do you want to continue? (Y/N)
```

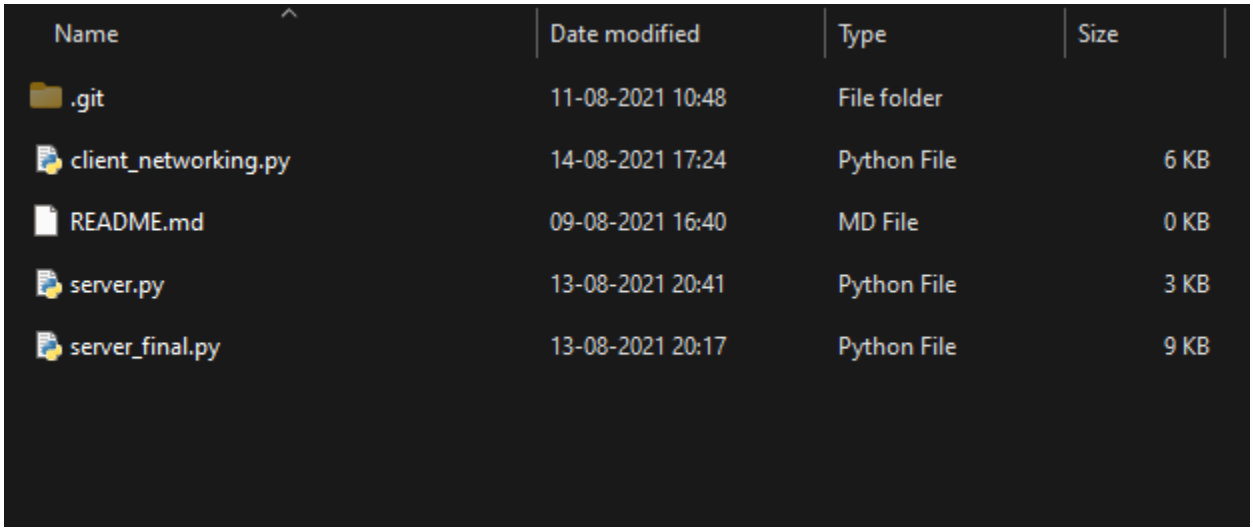
```
PS E:\Projects\Semester5\Networking\Project> & C:/Users/nirma/AppData/Local/Microsoft/WindowsApps/python3.9.exe e:/Projects/Semester5/Networking/Project/server.py
Server starting up.
Server address:  ('DESKTOP-H35TRG8', 5001)
New connection established with ('192.168.56.1', 7416)
[ 17:16:28 ] Operation  Enter  has been initiated
Values recieved:
{'Usage': 11915, 'allocation': 12000, 'expusage': 11900, 'region': 'Southwest', 'year': '12-08-2021'}
[ 17:16:51 ] Operation  Enter  has been successfully executed
```

```
Choose operation:
Inspection (I)
Type (T)
List by region (R)
I
Enter values to be inserted:
Region: Southwest
Length of cabling: 10000
Type of cable: cat-4
Cost of installation: 12000000
Date of installation: 12-08-2019
Number of issues reported: 2
Last date of inspection: 12-08-2020
Next check date: 12-08-2021
Status of sending:  183
Do you want to continue? (Y/N)
```

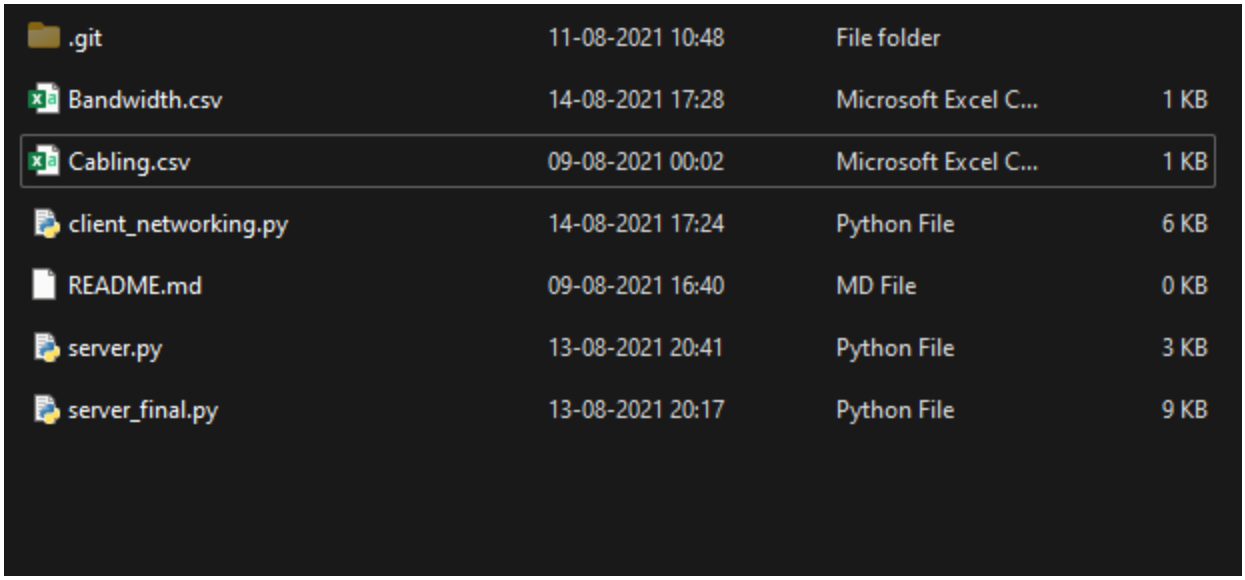
```
Bandwidth (B)
Cabling statistics (C)
Check server status (S)
S
Google: 142.250.77.164
None : Status of connection
PS E:\Projects\Semester5\Networking>
```

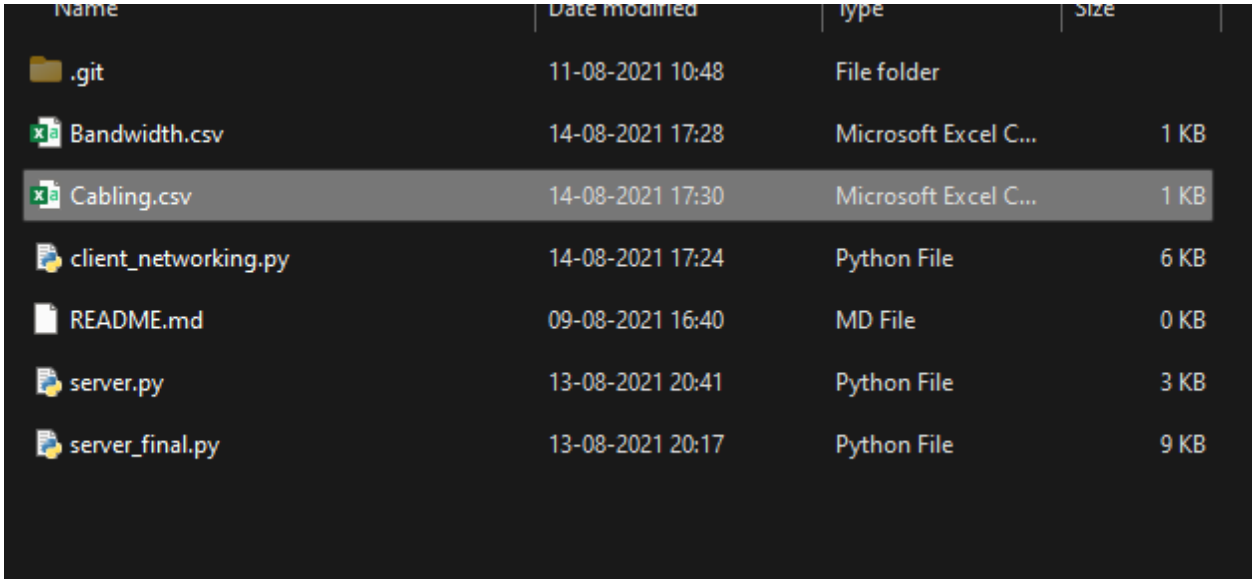
```
B
Bandwidth
Enter data (E)
View by region [V]
Check overload (O)
Enter your choice: V
         year     region  allocation  expusage  Usage
0  28-08-2020  Northwest         100        95    102
1  28-02-2020  Southwest        1200      1295   1300
2  31-03-2021  Northeast        1400      1209   1300
Do you want to continue? (Y/N)
```

```
[ 17:16:51 ] Operation  Enter  has been successfully executed
New connection established with ('192.168.56.1', 2049)
New connection established with ('192.168.56.1', 14179)
[ 17:23:36 ] Operation  View  has been initiated
[ 17:23:36 ] Operation  View  has been successfully executed
```



Name	Date modified	Type	Size
.git	11-08-2021 10:48	File folder	
client_networking.py	14-08-2021 17:24	Python File	6 KB
README.md	09-08-2021 16:40	MD File	0 KB
server.py	13-08-2021 20:41	Python File	3 KB
server_final.py	13-08-2021 20:17	Python File	9 KB


.git 11-08-2021 10:48 File folder
Bandwidth.csv 14-08-2021 17:28 Microsoft Excel C... 1 KB
Cabling.csv 09-08-2021 00:02 Microsoft Excel C... 1 KB
client_networking.py 14-08-2021 17:24 Python File 6 KB
README.md 09-08-2021 16:40 MD File 0 KB
server.py 13-08-2021 20:41 Python File 3 KB
server_final.py 13-08-2021 20:17 Python File 9 KB


Name Date modified Type Size
.git 11-08-2021 10:48 File folder
Bandwidth.csv 14-08-2021 17:28 Microsoft Excel C... 1 KB
Cabling.csv 14-08-2021 17:30 Microsoft Excel C... 1 KB
client_networking.py 14-08-2021 17:24 Python File 6 KB
README.md 09-08-2021 16:40 MD File 0 KB
server.py 13-08-2021 20:41 Python File 3 KB
server_final.py 13-08-2021 20:17 Python File 9 KB

INTERNET SERVICE PROVIDER (ISP)
CSE301 – COMPUTER NETWORKS

Roll .No	Department	Purpose	Network Details
023	Marketing team	Promoting the business. Responsible for creating and disseminating images, messages and ideas that best communicate the company value	**No. of end nodes:36** **No. of servers: 5** **Protocols:** **HTTP, FTP, DNS, POP3, SMTP, TCP/IP,RIP, OSPF**
025	Customer care	Handling feedback and complaints from customers. Responsible for establishing the relationship between the company and the public.	
038	Network Team	Provide networking services to the other employees, customers and troubleshoot any problem that arises related to connections.	

MARKETING TEAM

170.18.10.x:

IP Class: B
Default Subnet Mask: 255.255.0.0 - /16 - 11111111.11111111.**00000000.00000000**

Subnet Mask: 255.255.248 - **11111111.11111111.11111111.11111**000 - /29

Wildcard Mask: 0.0.0.7 (Inverse of subnet mask(255-248))
No. of host bits: 3
No. of Network bits: 29-16 = 13
No. of usable hosts in each subnet: 2^3 -2 = 6
No.of subnets: 2^(29-16) = 2^13 = 8192
IP Class: C

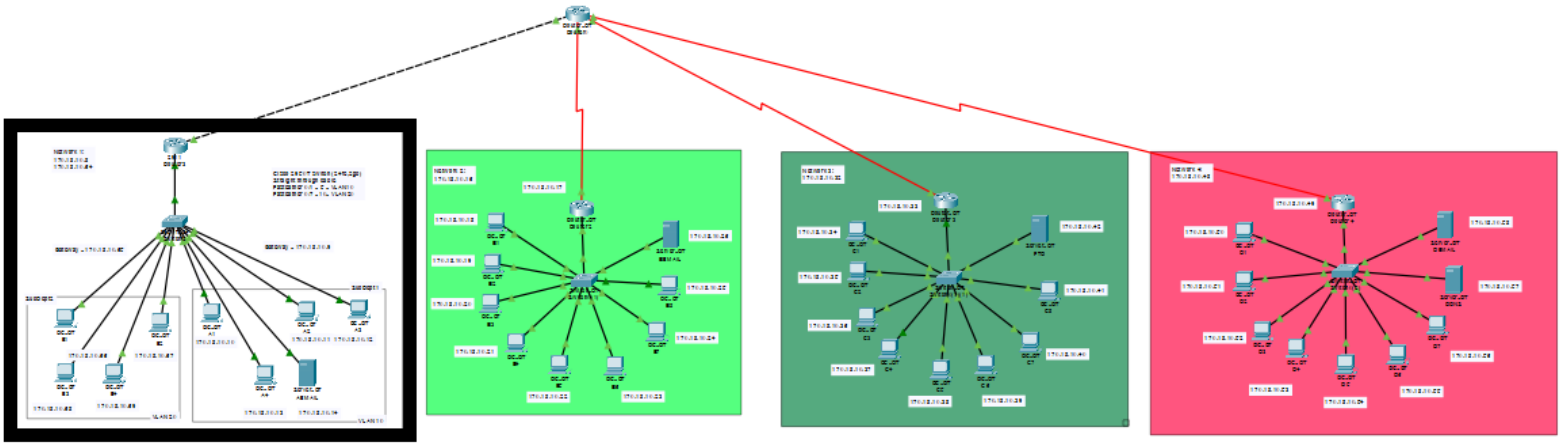
255.255.240 - **11111111.11111111.11111111.1111**0000 - /28

Wildcard Mask: 0.0.0.15 (Inverse of subnet mask(255-240))
No. of host bits: 4
No. of Network bits: 28-16 = 12
No. of usable hosts in each subnet: 2^4 -2 = 14
No.of subnets: 2^(28-16) = 2^12 = 4096
IP Class: C

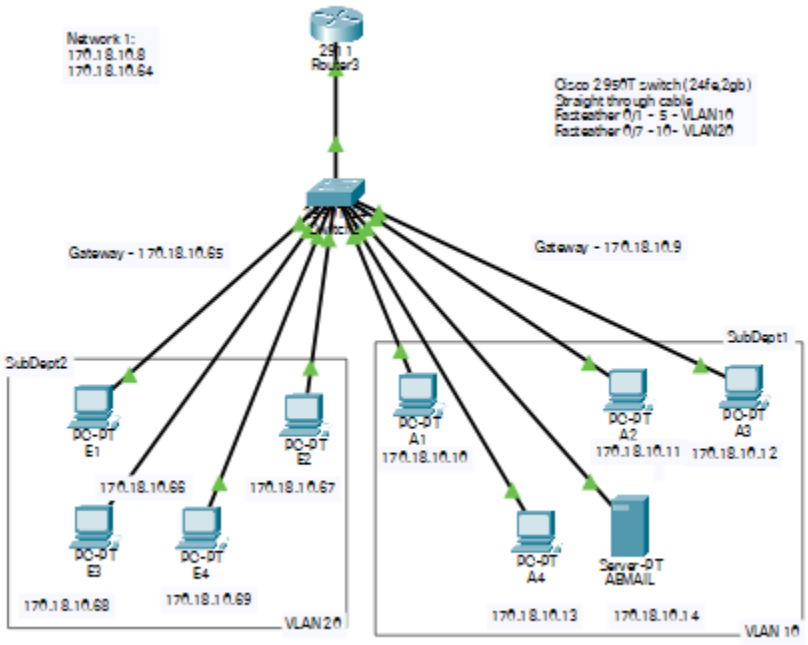
IP Scheming:

Department	Subnet Mask	Wildcard Mask	Network Address	Gateway	Host IP addresses	Broadcast Address	Number of usable addresses
1	255.255.248	0.0.0.7	170.18.10.8	170.18.10.9	170.18.10.10 -A1 170.18.10.11 - A2 170.18.10.12 - A3 170.18.10.13 - A4 170.18.10.14 - ES	170.18.10.15	6
			170.18.10.64	170.18.10.65	170.18.10.66 - E1 170.18.10.67 - E2 170.18.10.68 - E3 170.18.10.69 - E4	170.18.10.71	
2	255.255.240	0.0.0.15	170.18.10.16	170.18.10.17	170.18.10.18 - B1 170.18.10.19 - B2 170.18.10.20 - B3 170.18.10.21 - B4 170.18.10.22 - B5 170.18.10.23 - B6 170.18.10.24 - B7 170.18.10.25 - B8	170.18.10.31	14

					170.18.10.26 - ES		
3	255.255.240	0.0.0.15	170.18.10.32	170.18.10.33	170.18.10.34 - C1 170.18.10.35 - C2 170.18.10.36 - C3 170.18.10.37 - C4 170.18.10.38 - C5 170.18.10.39 - C6 170.18.10.40 - C7 170.18.10.41 - C8 170.18.10.42 - FTP	170.18.10.47	14
4	255.255.240	0.0.0.15	170.18.10.48	170.18.10.49	170.18.10.50 - D1 170.18.10.51 - D2 170.18.10.52 - D3 170.18.10.53 - D4 170.18.10.54 - D5 170.18.10.55 - D6 170.18.10.56 - D7 170.18.10.57 - DNS 170.18.10.58 - WEB	170.18.10.63	14

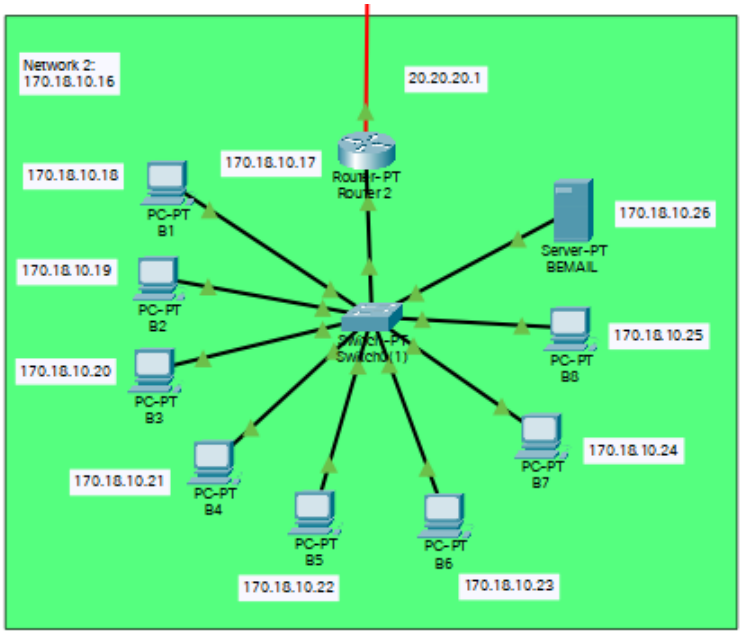


Department 1:

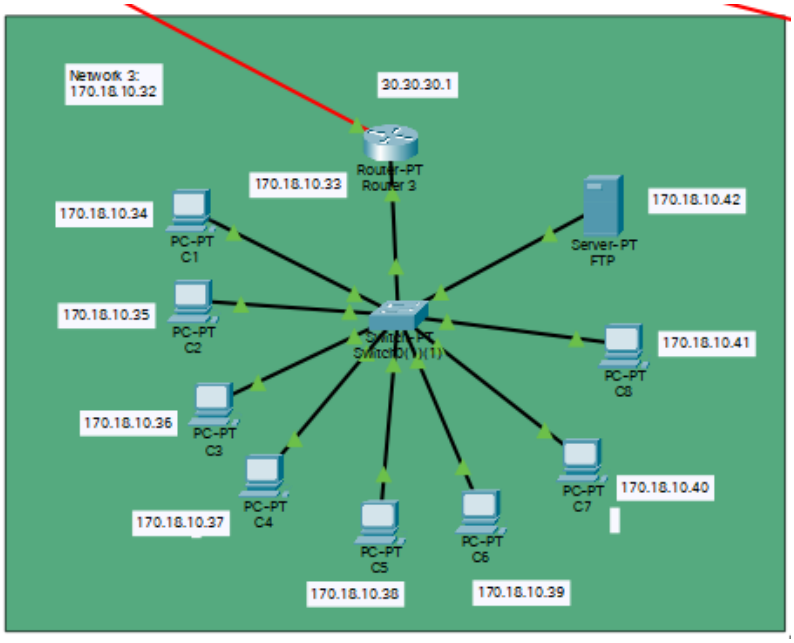


Fastether 0/1 - 5 - VLAN10
Fastether 0/7 -10- VLAN20

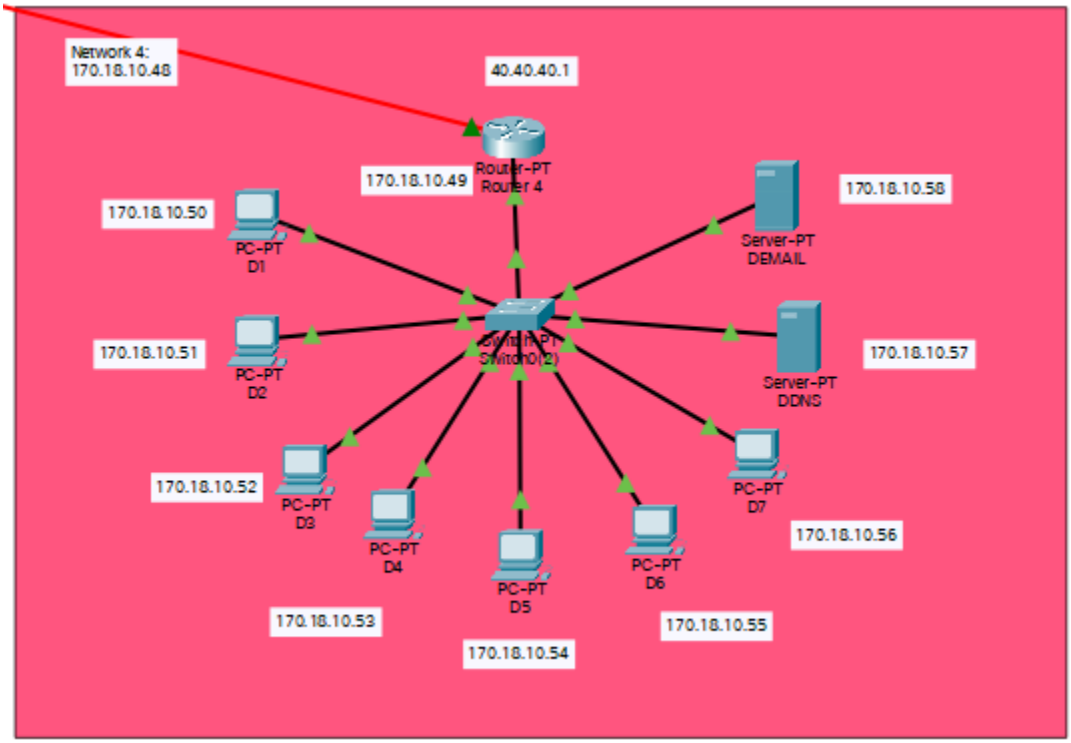
Department 2:



Department 3:



Department 4:



VLAN

>> This case study contains 4 departments where **Department 1 contains VLAN**

Configuring ports:

#SWITCH

```
Switch#show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/6, Fa0/11, Fa0/12, Fa0/13
                                                Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                                Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                                Fa0/22, Fa0/23, Fa0/24, Gig0/1
                                                Gig0/2
10   ***SubDept1***                   active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5
20   ***SubDept2***                   active    Fa0/7, Fa0/8, Fa0/9, Fa0/10
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

VLAN10:

```
interface FastEthernet0/1
 switchport access vlan 10
 switchport mode access
!
interface FastEthernet0/2
 switchport access vlan 10
 switchport mode access
!
interface FastEthernet0/3
 switchport access vlan 10
 switchport mode access
!
interface FastEthernet0/4
 switchport access vlan 10
 switchport mode access
!
interface FastEthernet0/5
 switchport access vlan 10
 switchport mode access
!
```

VLAN20:

```
interface FastEthernet0/7
 switchport access vlan 20
 switchport mode access
!
interface FastEthernet0/8
 switchport access vlan 20
 switchport mode access
!
interface FastEthernet0/9
 switchport access vlan 20
 switchport mode access
!
interface FastEthernet0/10
 switchport access vlan 20
 switchport mode access
!
```

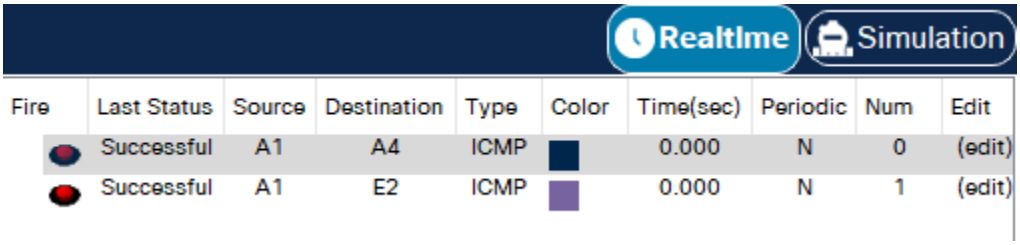
SW-RT:

```
interface GigabitEthernet0/1
 switchport mode trunk
!
```

VLAN Gateways:

```
Router#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     unassigned      YES unset  up                    up
GigabitEthernet0/0.10  170.18.10.9     YES manual up                    up
GigabitEthernet0/0.20  170.18.10.65    YES manual up                    up
GigabitEthernet0/1     unassigned      YES unset  administratively down down
GigabitEthernet0/2     unassigned      YES unset  administratively down down
Vlan1                  unassigned      YES unset  administratively down down
Router#
```

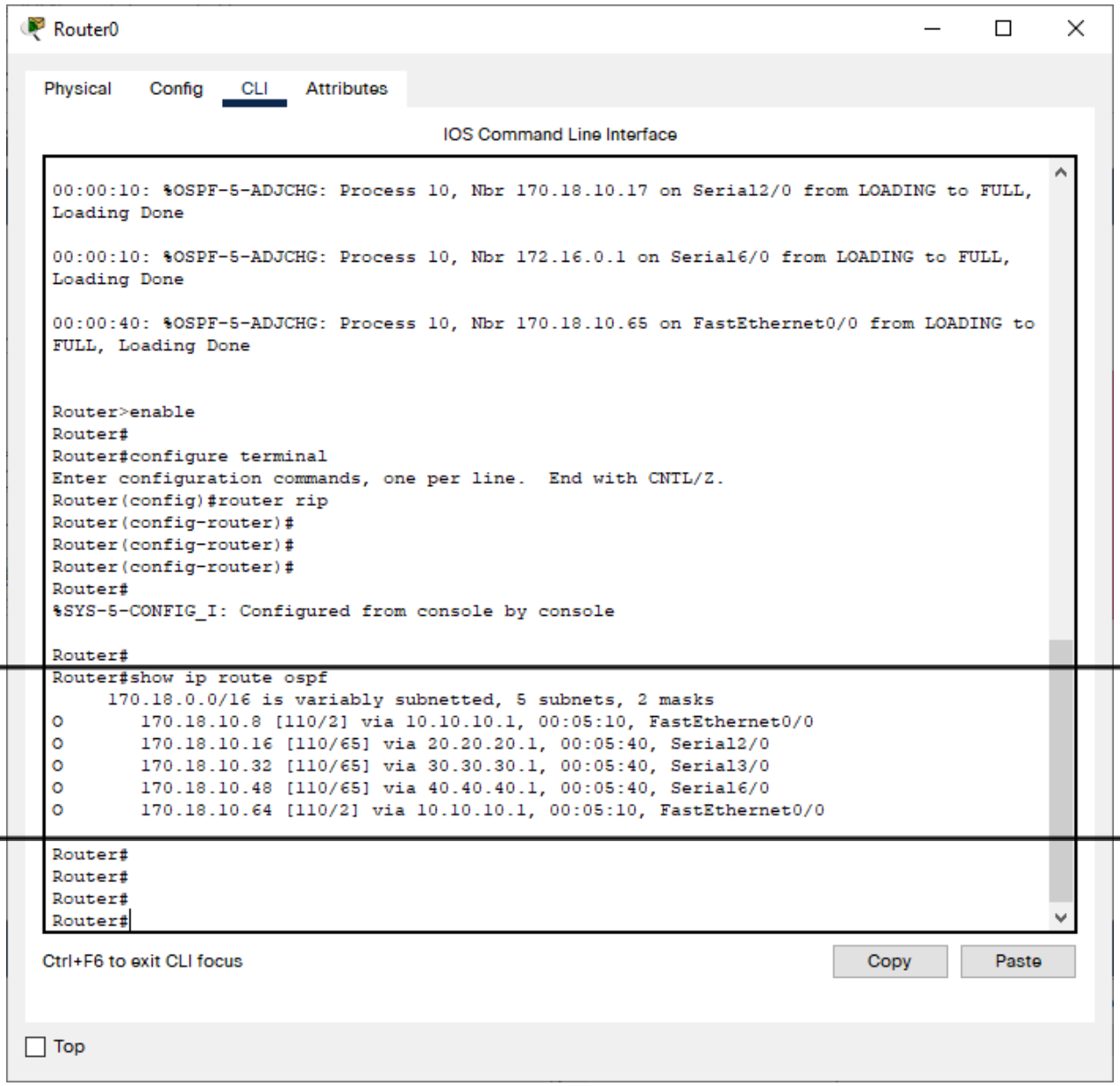
Communication within a VLAN and between 2 VLAN:



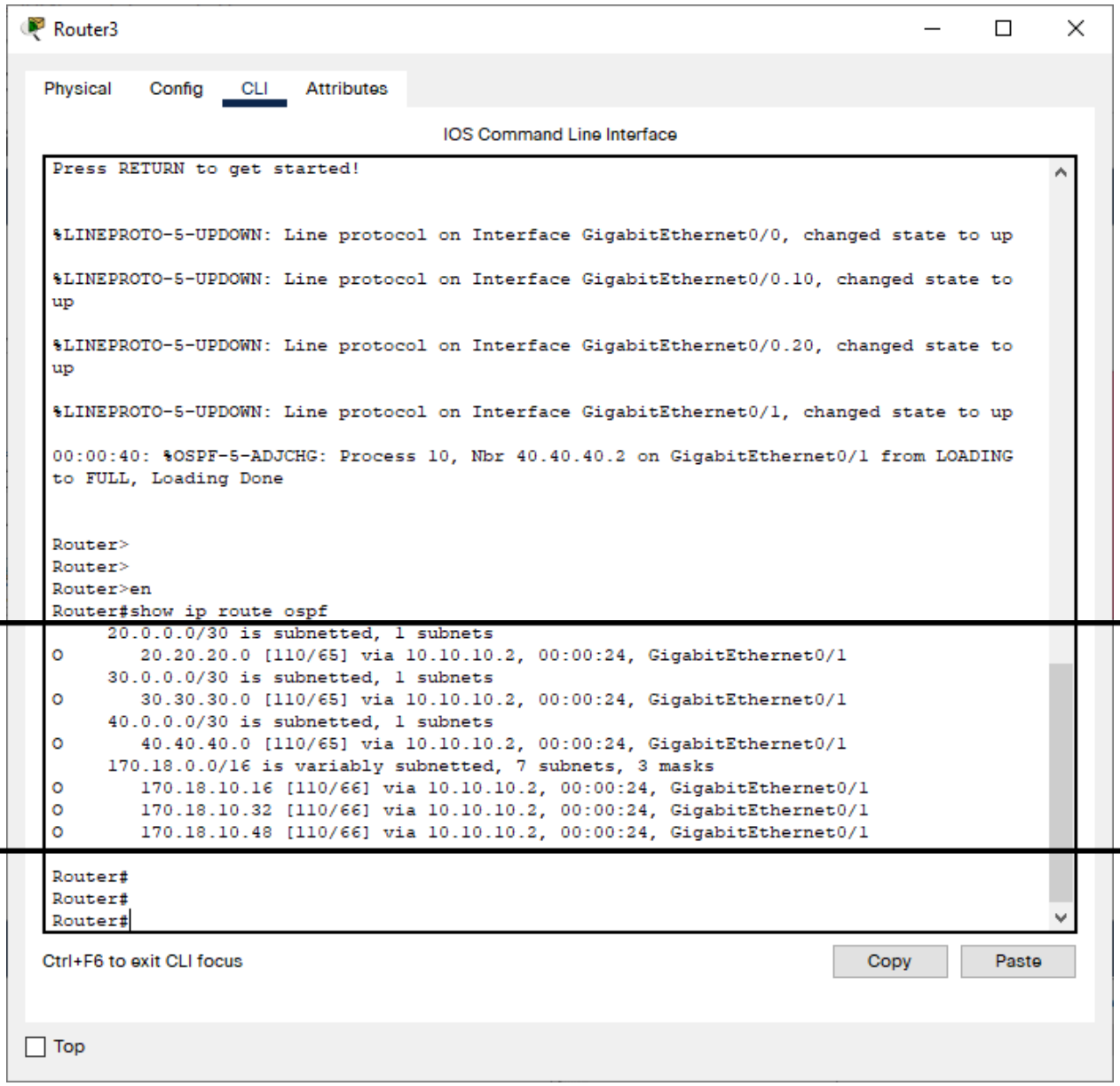
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit
	Successful	A1	A4	ICMP		0.000	N	0	(edit)
	Successful	A1	E2	ICMP		0.000	N	1	(edit)

OSPF:

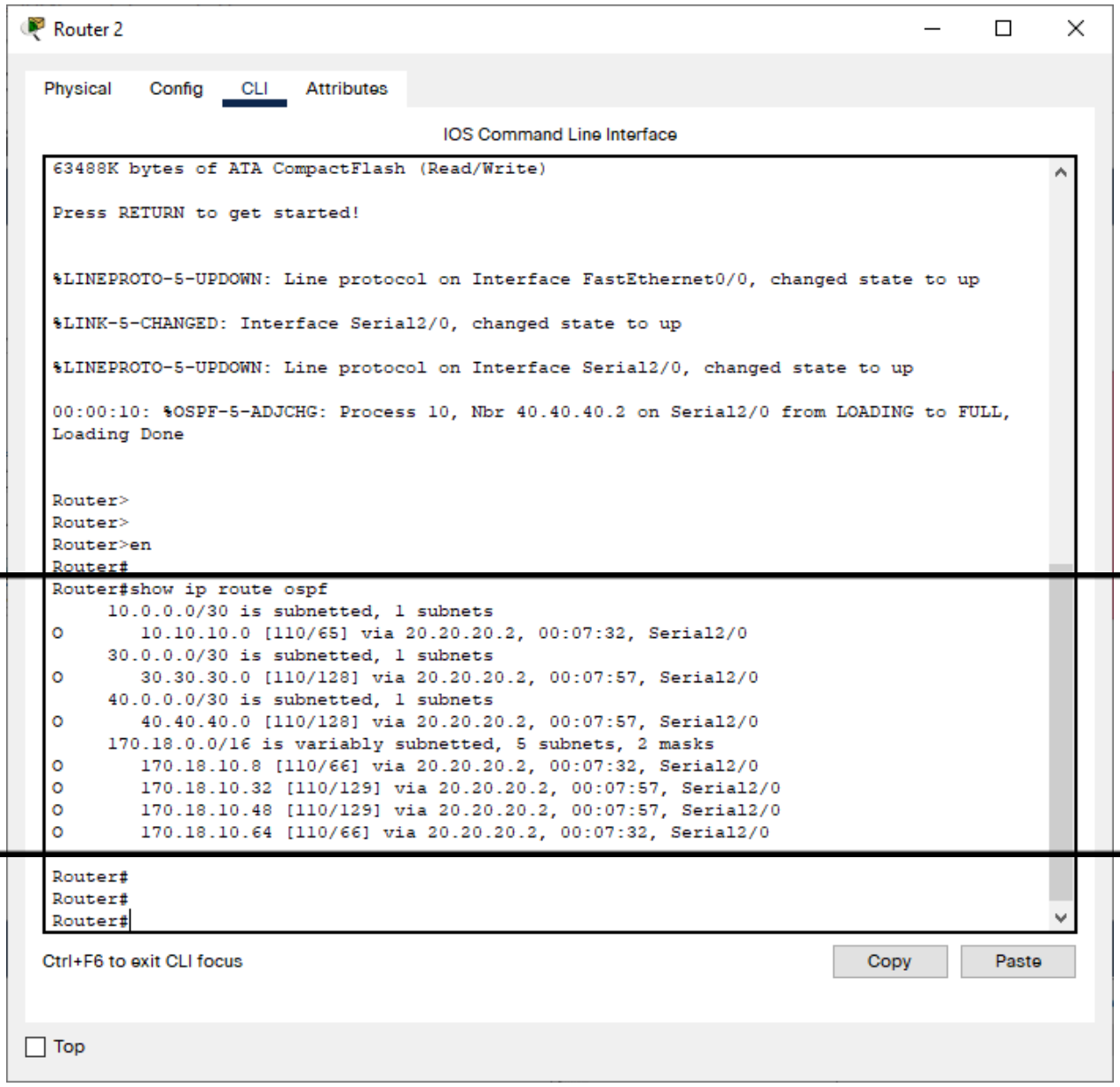
Router 0:



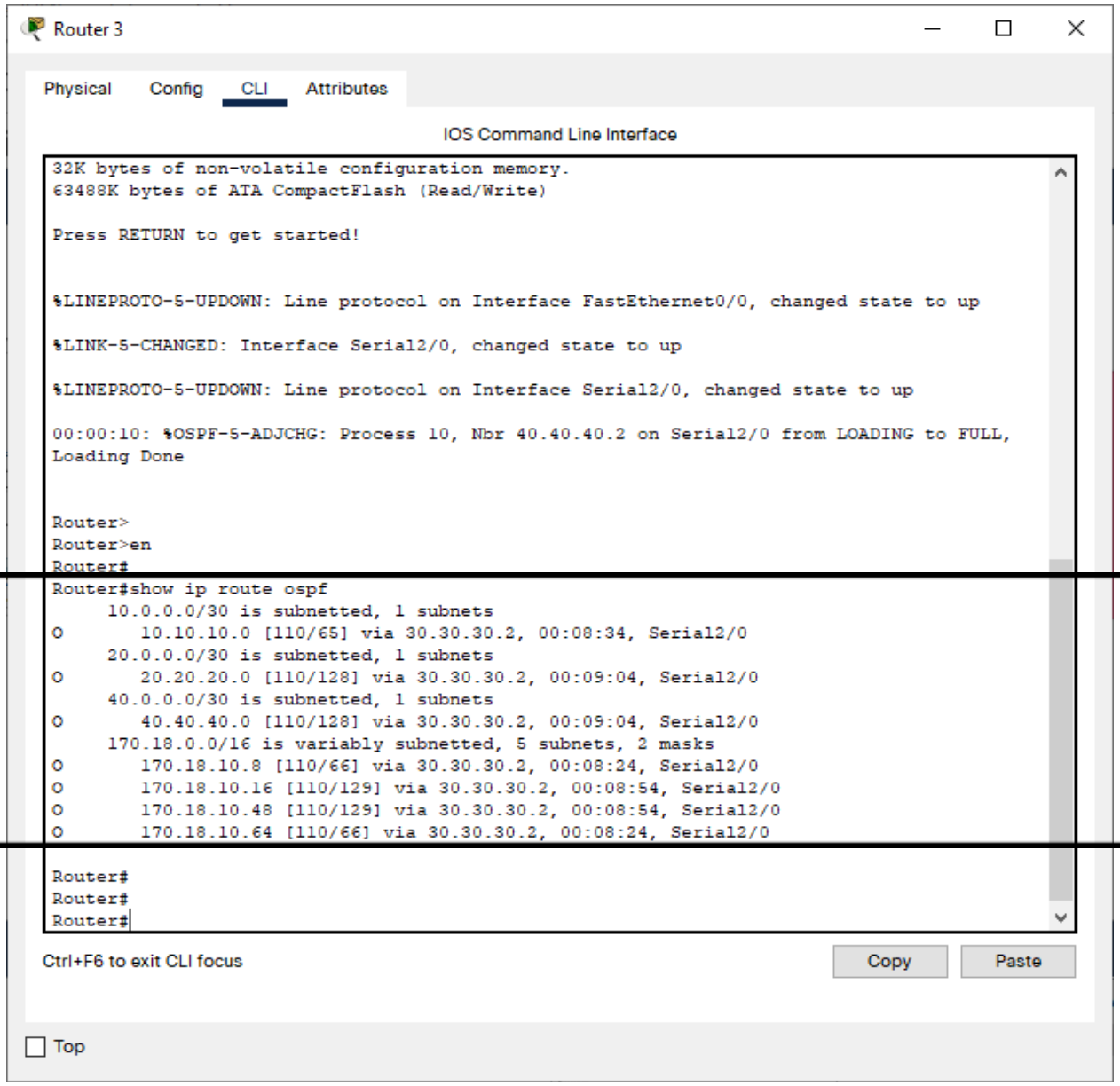
Router1: (Department 1)



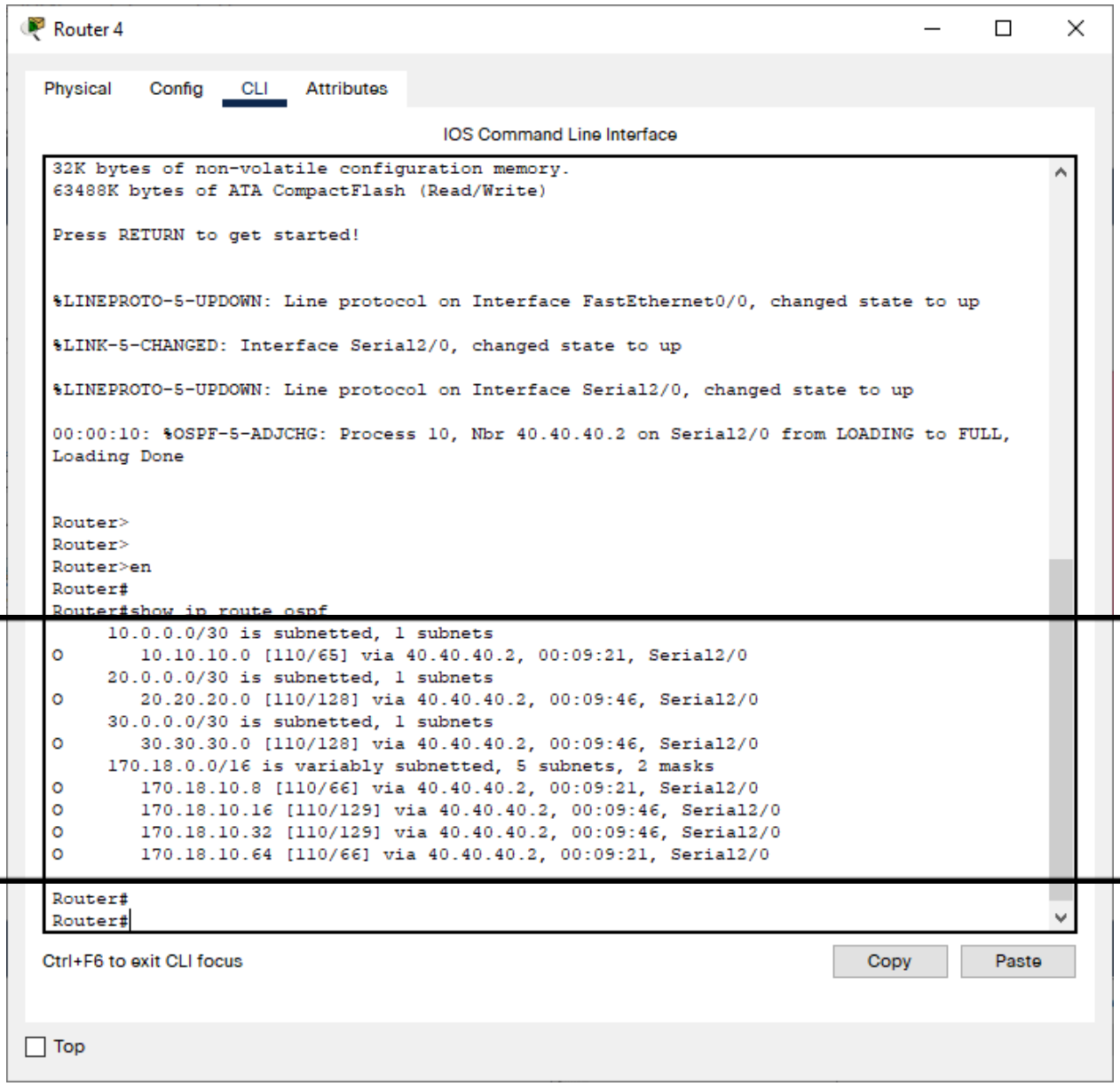
Router 2: (Department)



Router 3: (Department 3)



Router 4: (Department 4)



Communication between Different departments using OSPF routing:

Fire	Last Status	Source	Destination	Type	Color	Time(sec)
	Successful	B3	E1	ICMP		0.000
	Successful	E4	B2	ICMP		0.000
	Successful	A2	B5	ICMP		0.000

Fire	Last Status	Source	Destination	Type	Color	Time(sec)
	Successful	C8	D2	ICMP		0.000
	Successful	C3	B8	ICMP		0.000
	Successful	D2	B8	ICMP		0.000

RIP - V1: (Default Version)

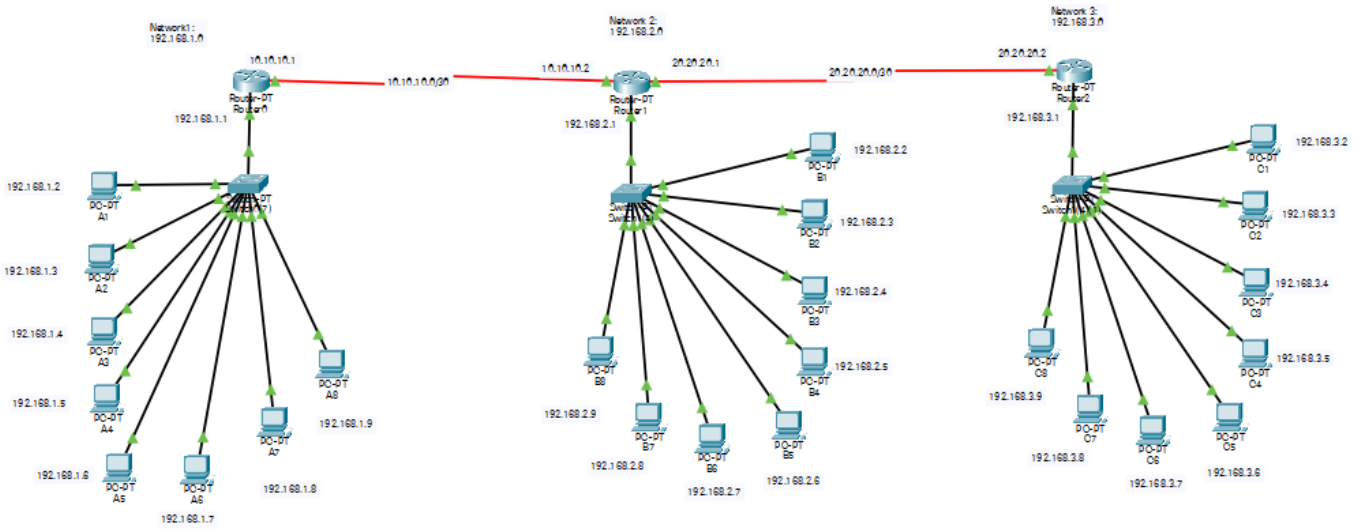
The Gateway addresses (Routers):
170.18.10.9/29
170.18.10.17/28
170.18.10.33/28
170.18.10.49/28
170.18.10.65/29
Belongs to class B in classful addressing and advertises each other with their default subnet mask (/16)
When a router learns multiple routes to a specific network via multiple routing protocols, it installs the route with the lowest distance in the routing table.
If the router receives and installs multiple paths with the same distance, **load-balancing** can occur.

CLASSFUL RP	**CLASSLESS RP**
• RIP V1	• RIP V2 • OSPF

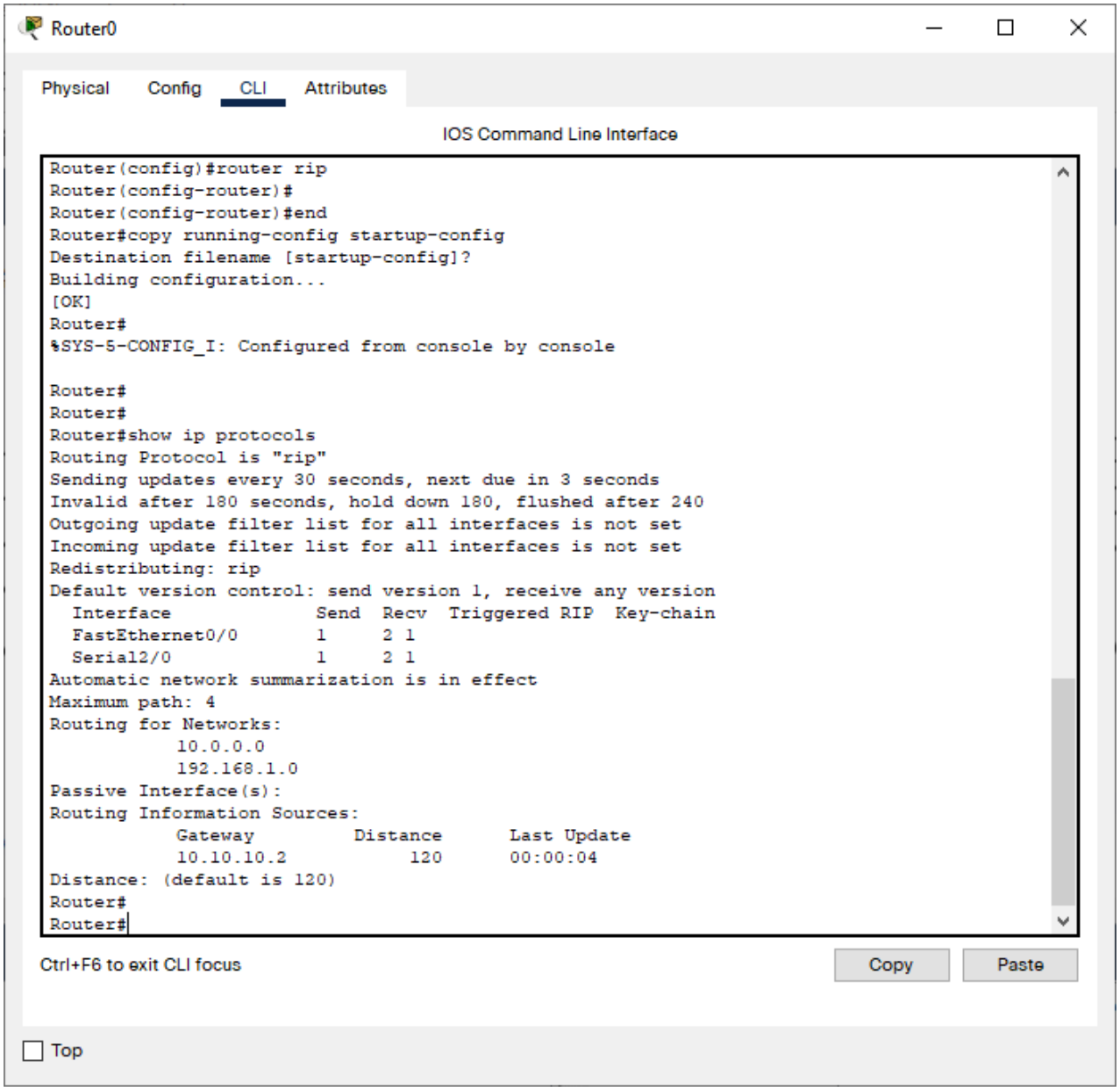
192.168.a.x: (a:1,2,3)
IP CLASS: C
Default Subnet Mask: 255.255.255.0 - /24 - 11111111.11111111.11111111.**00000000**
Wildcard Mask: 0.0.0.255 (Inverse of subnet mask)
Usable Host IP range: 192.168.a.1 - 192.168.a.254
No. of host bits: 8
No. of Network bits:24
No. of usable hosts: 2^8 -2 =254

Network	Network Address	Gateway	Host IP addresses
1	192.168.1.0	192.168.1.1	192.168.1.2 - 9
2	192.168.2.0	192.168.2.1	192.168.2.2 - 9
3	192.168.3.0	192.168.3.1	192.168.3.2 - 9

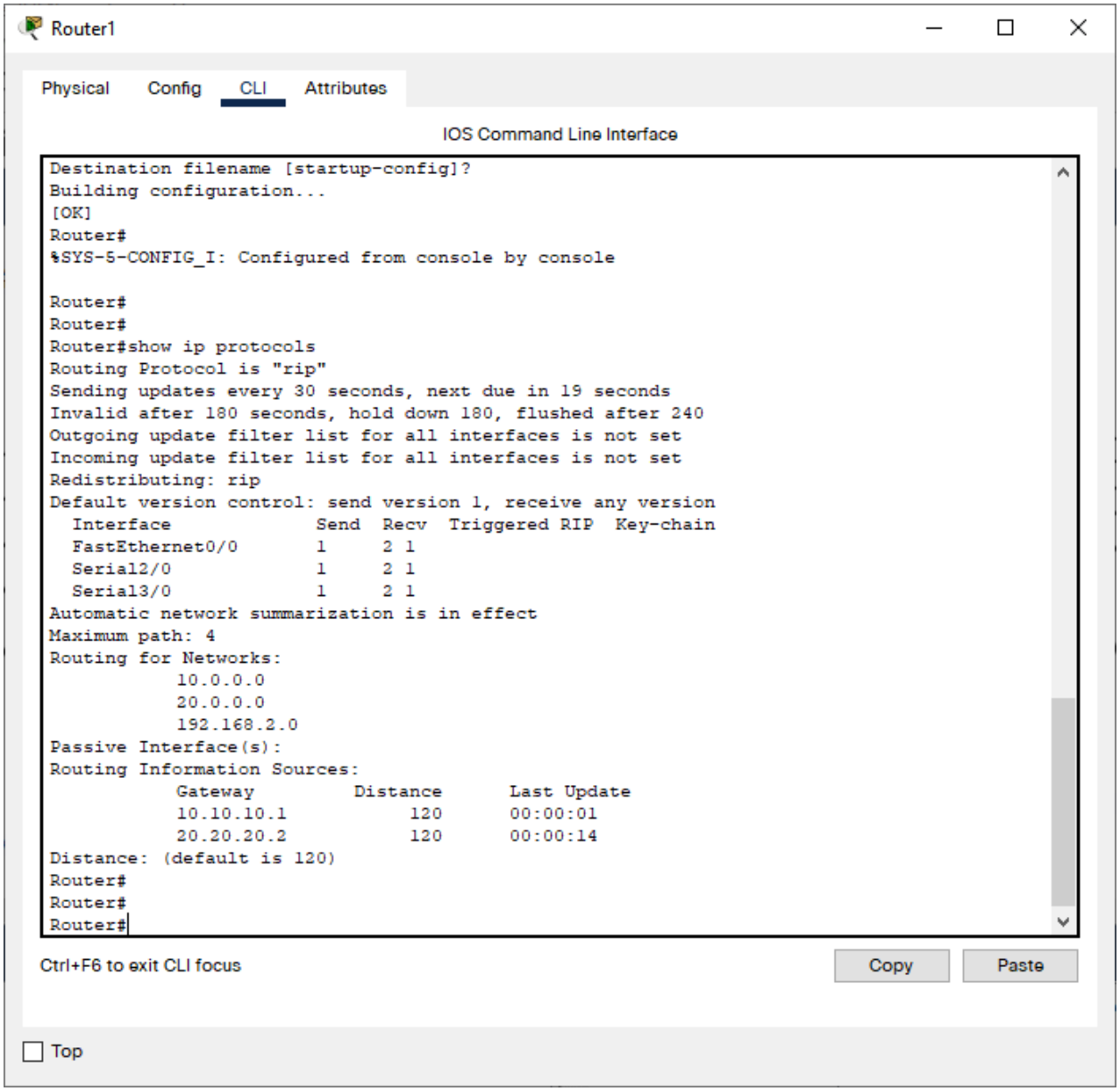
Network Model:



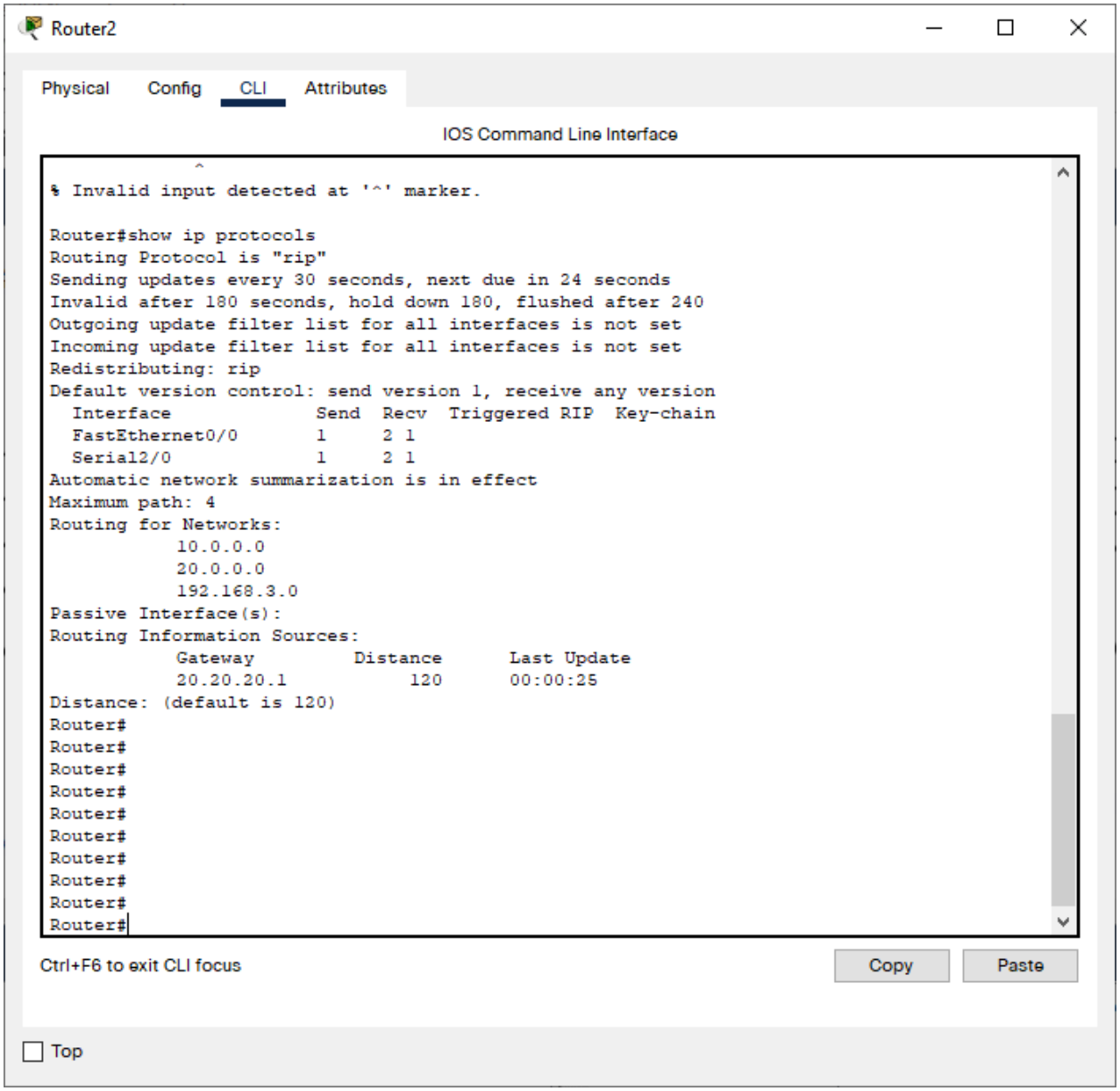
Router0:


Router0
Physical
Config
CLI
Attributes
IOS Command Line Interface
Router(config)#router rip
Router(config-router)#
Router(config-router)#end
Router#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#
Router#
Router#show ip protocols
Routing Protocol is "rip"
Sending updates every 30 seconds, next due in 3 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Redistributing: rip
Default version control: send version 1, receive any version
Interface Send Recv Triggered RIP Key-chain
FastEthernet0/0 1 2 1
Serial2/0 1 2 1
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
10.0.0.0
192.168.1.0
Passive Interface(s):
Routing Information Sources:
Gateway Distance Last Update
10.10.10.2 120 00:00:04
Distance: (default is 120)
Router#
Router#
Ctrl+F6 to exit CLI focus
Copy
Paste
Top

Router 1:


Router1
Physical
Config
CLI
Attributes
IOS Command Line Interface
Destination filename [startup-config]?
Building configuration...
[OK]
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#
Router#
Router#show ip protocols
Routing Protocol is "rip"
Sending updates every 30 seconds, next due in 19 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Redistributing: rip
Default version control: send version 1, receive any version
Interface Send Recv Triggered RIP Key-chain
FastEthernet0/0 1 2 1
Serial2/0 1 2 1
Serial3/0 1 2 1
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
10.0.0.0
20.0.0.0
192.168.2.0
Passive Interface(s):
Routing Information Sources:
Gateway Distance Last Update
10.10.10.1 120 00:00:01
20.20.20.2 120 00:00:14
Distance: (default is 120)
Router#
Router#
Router#
Ctrl+F6 to exit CLI focus
Copy
Paste
Top

Router 2:



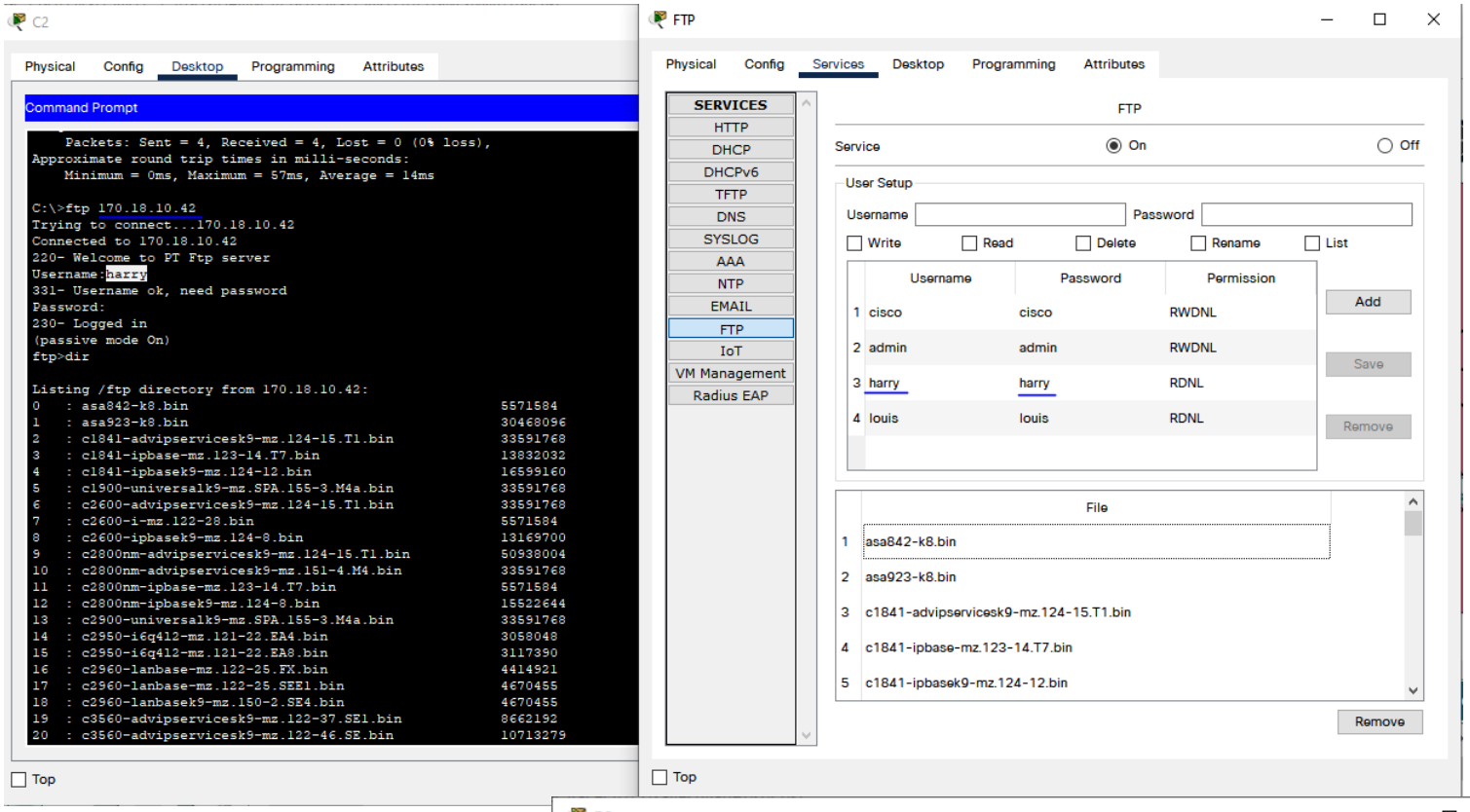
Communication using RIP:

Fire	Last Status	Source	Destination	Type	Color	Time(sec)
	Successful	A8	B8	ICMP		0.000
	Successful	B4	C8	ICMP		0.000
	Successful	C8	A8	ICMP		0.000

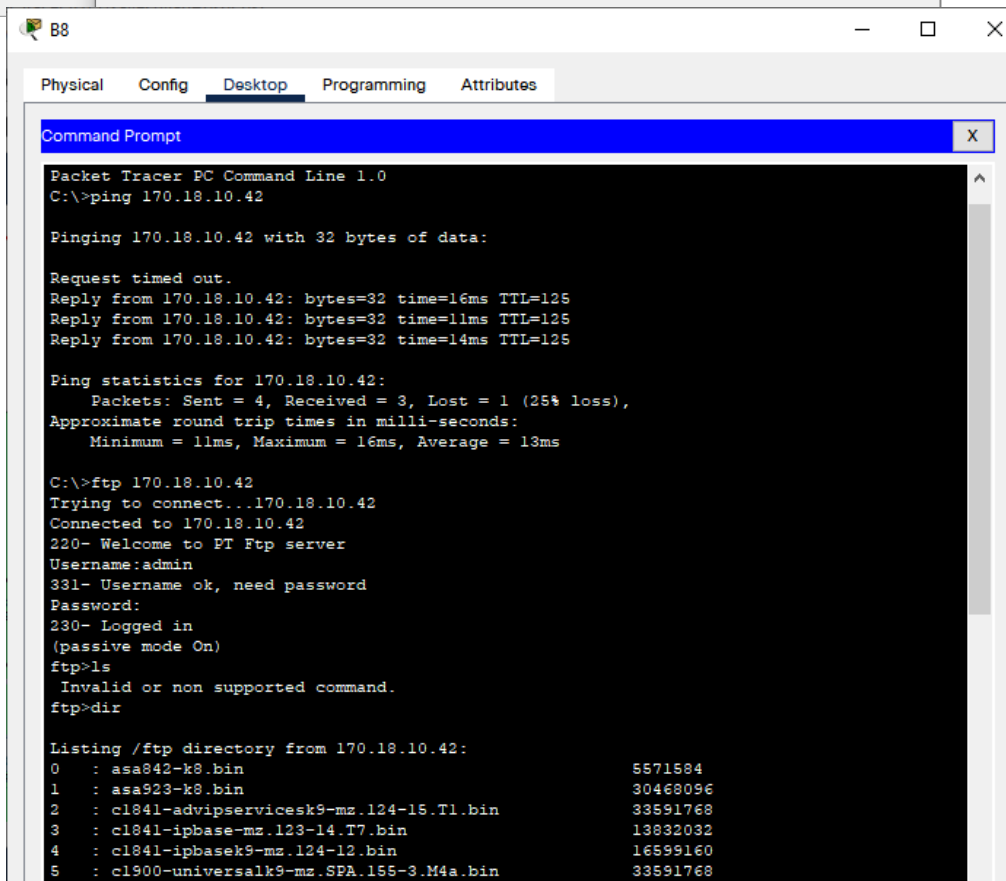
Application Layer Protocols:

>>FTP - File transfer Protocol:

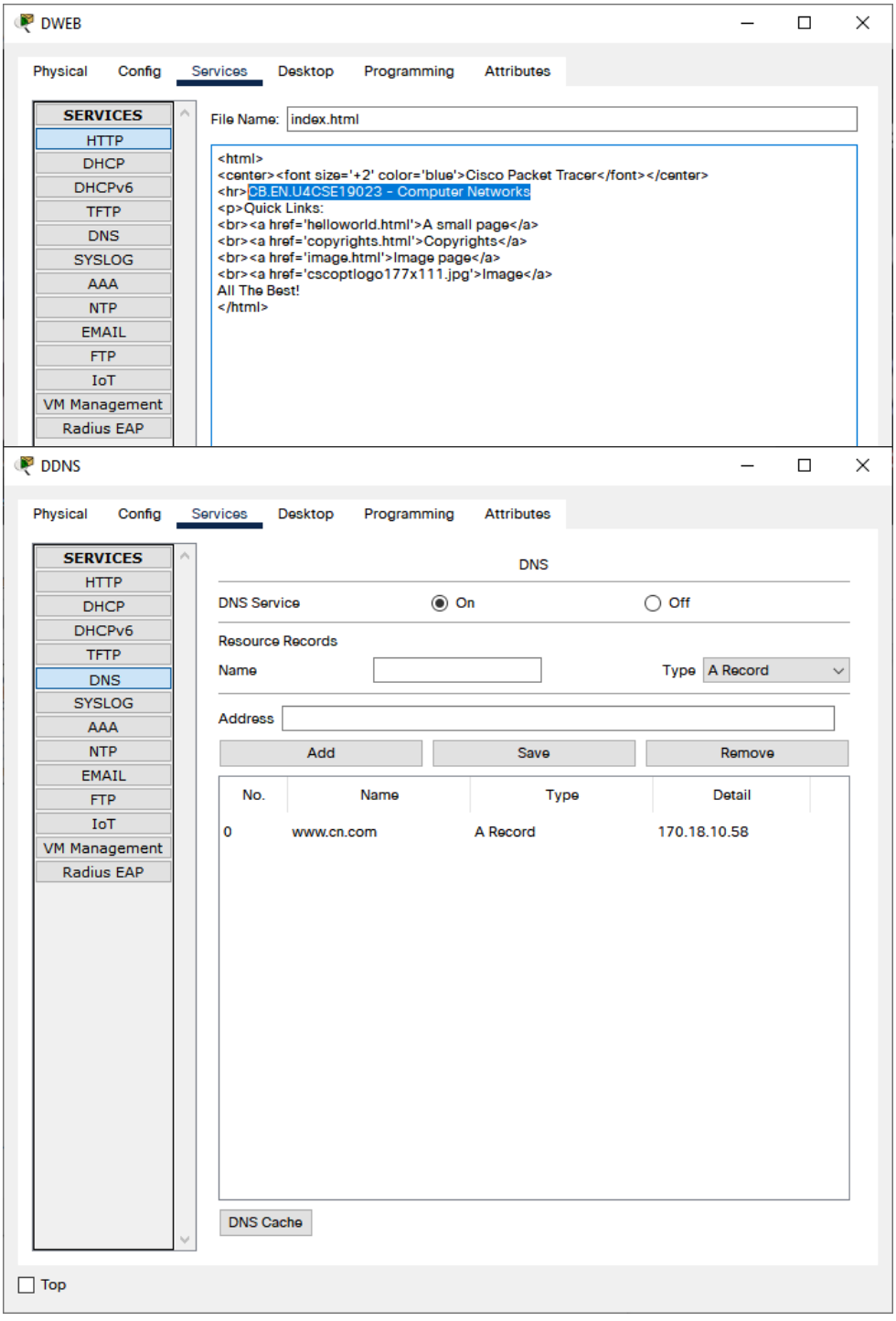
C2 - CFTP



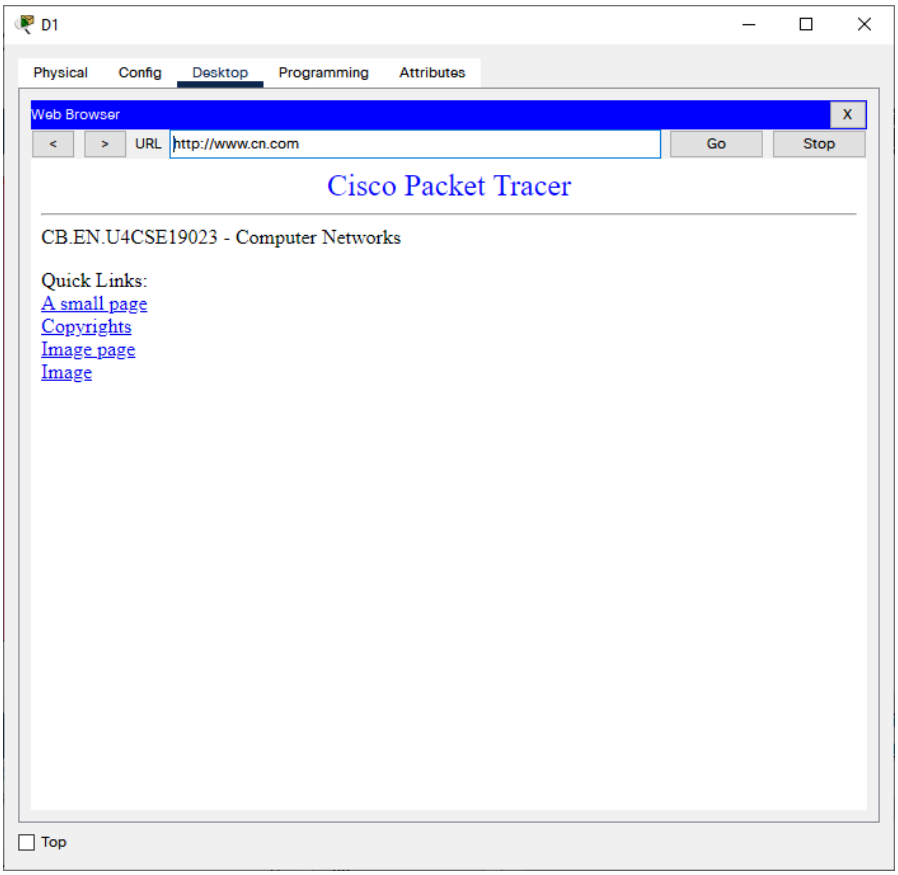
B8-CFTP



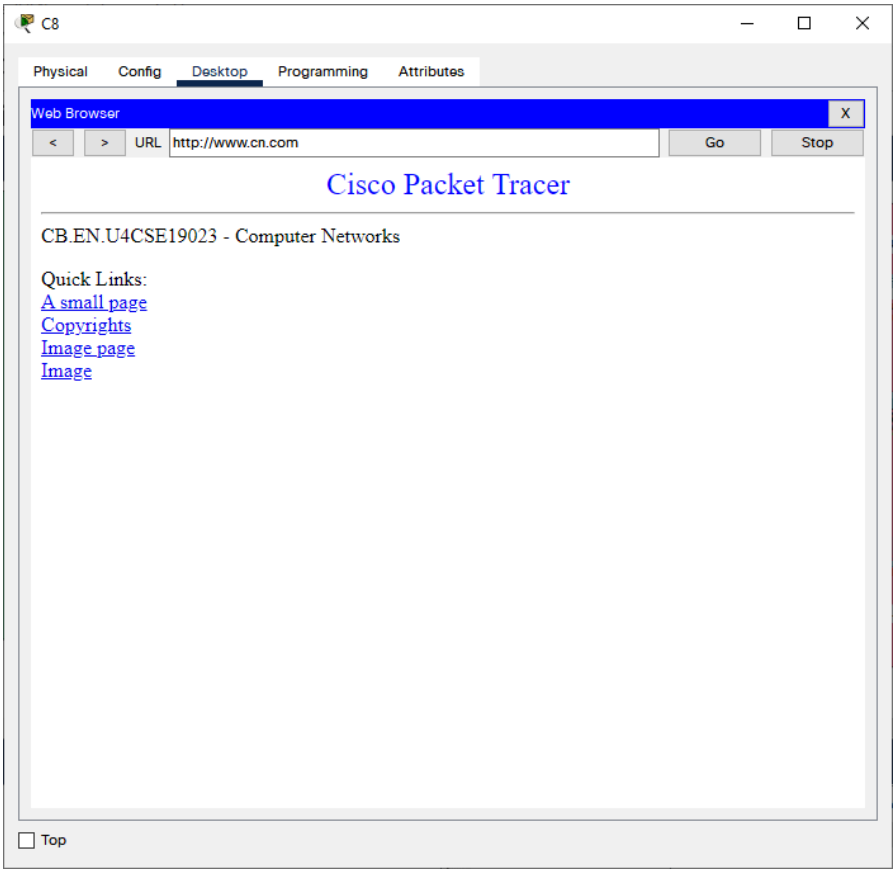
Web Server(HTTP) & DNS Server:



Accessing www.cn.com from D1 PC:

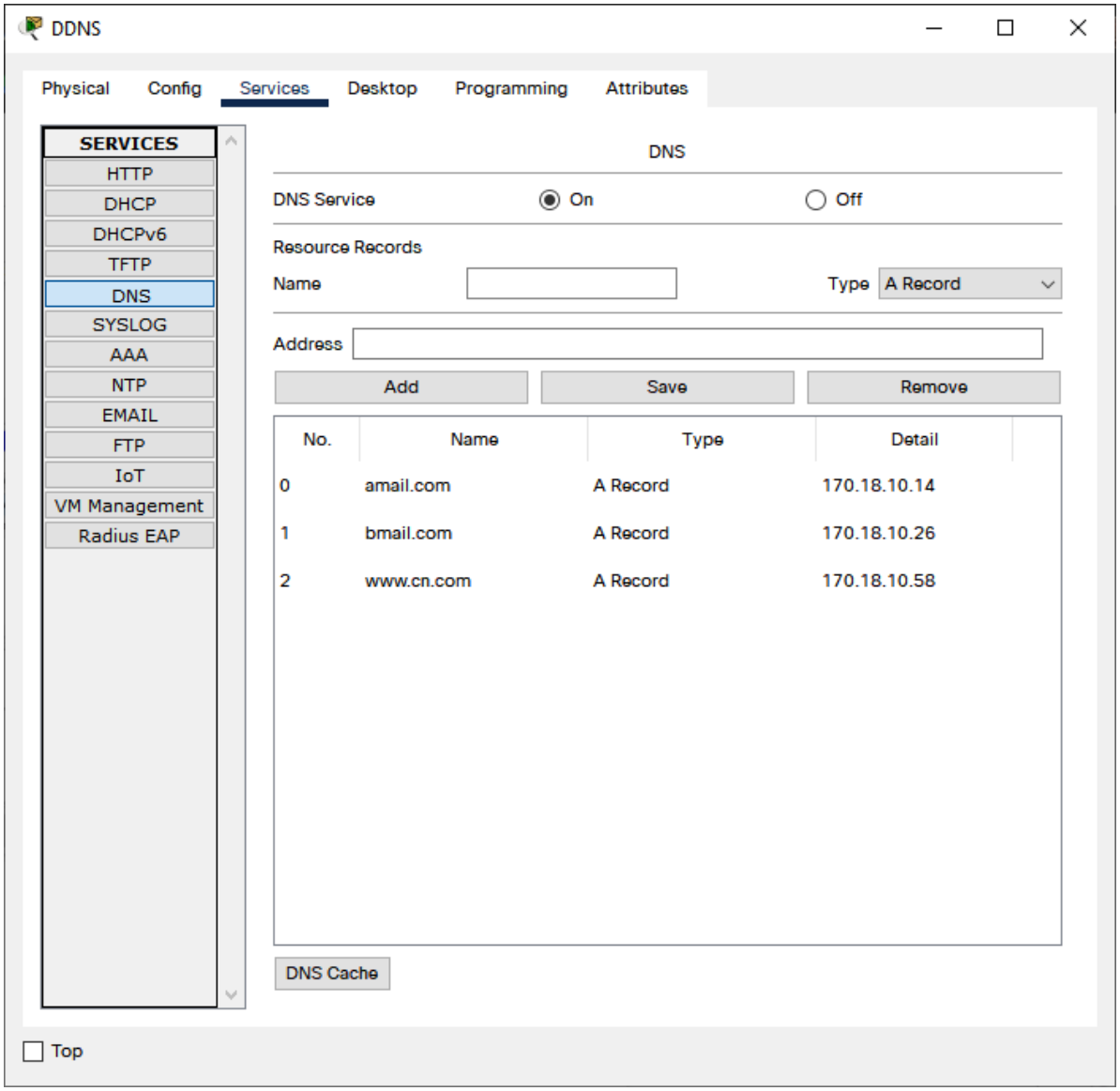


Accessing www.cn.com from C8 PC:

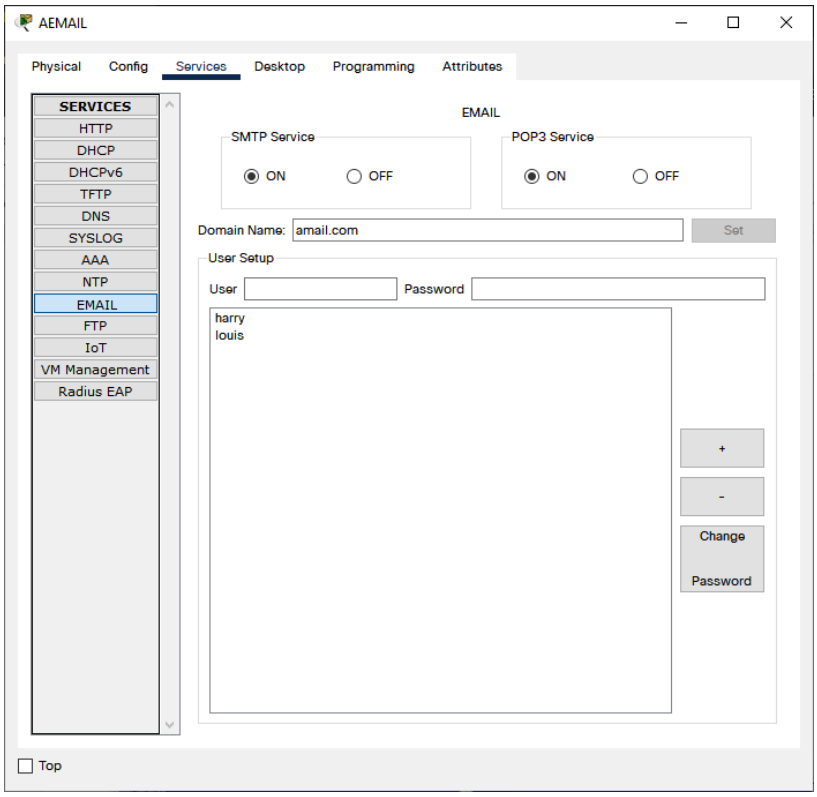


Email Server & DNS:

DNS:

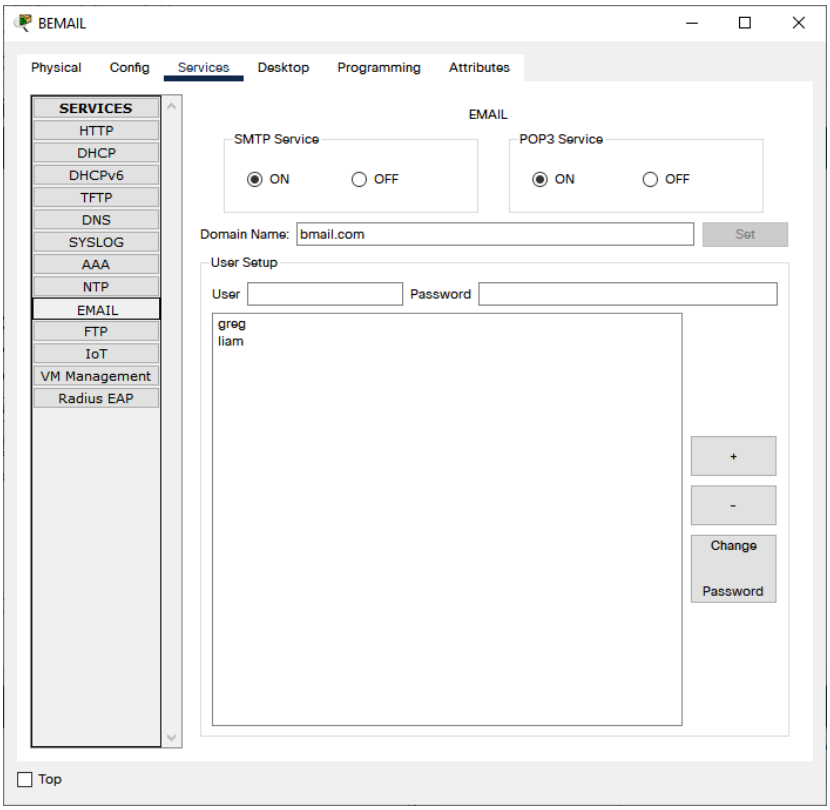


A - EMAIL SERVER:



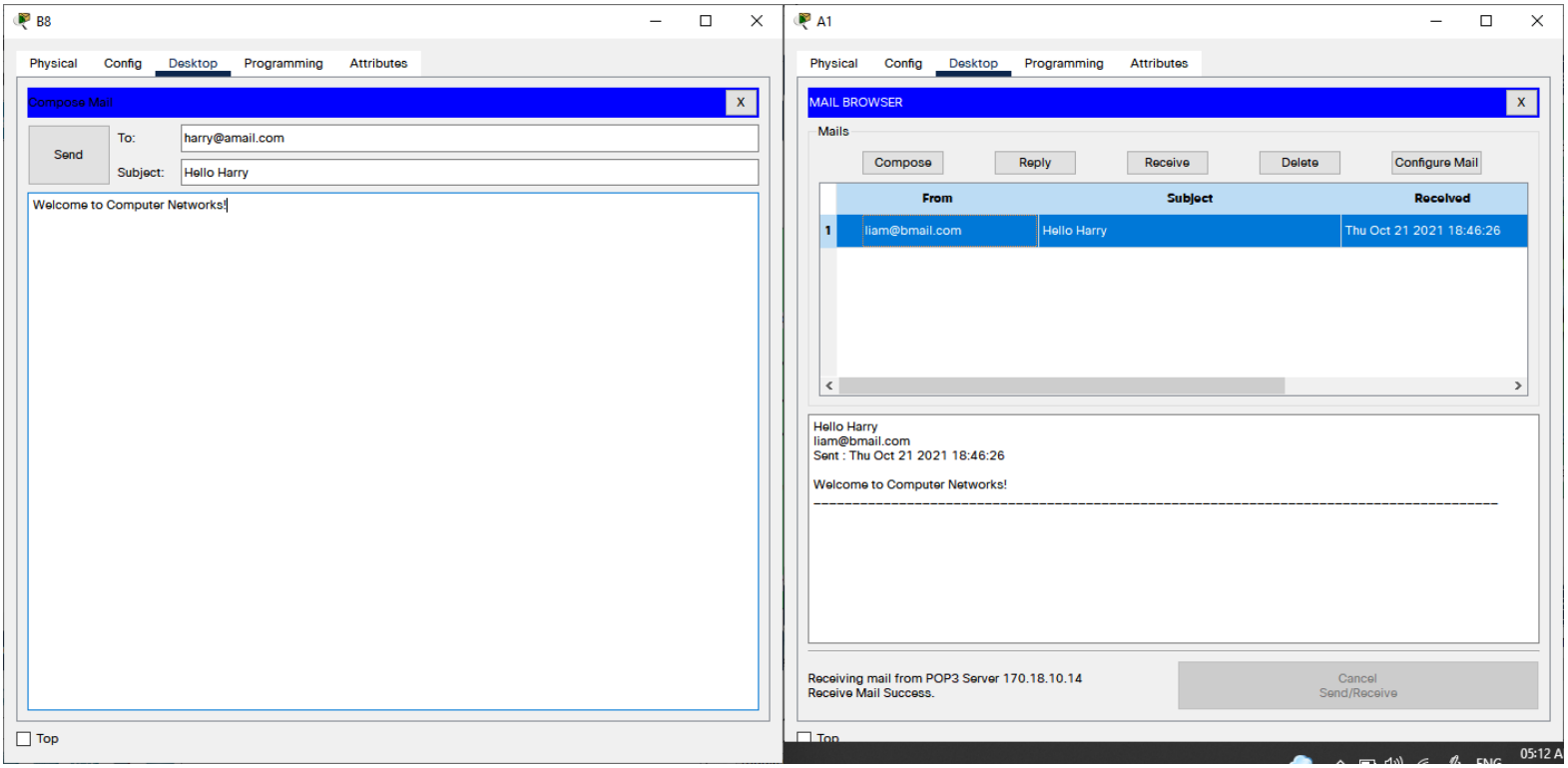
harry@amail.com
louis@amail.com

B - EMAIL SERVER:

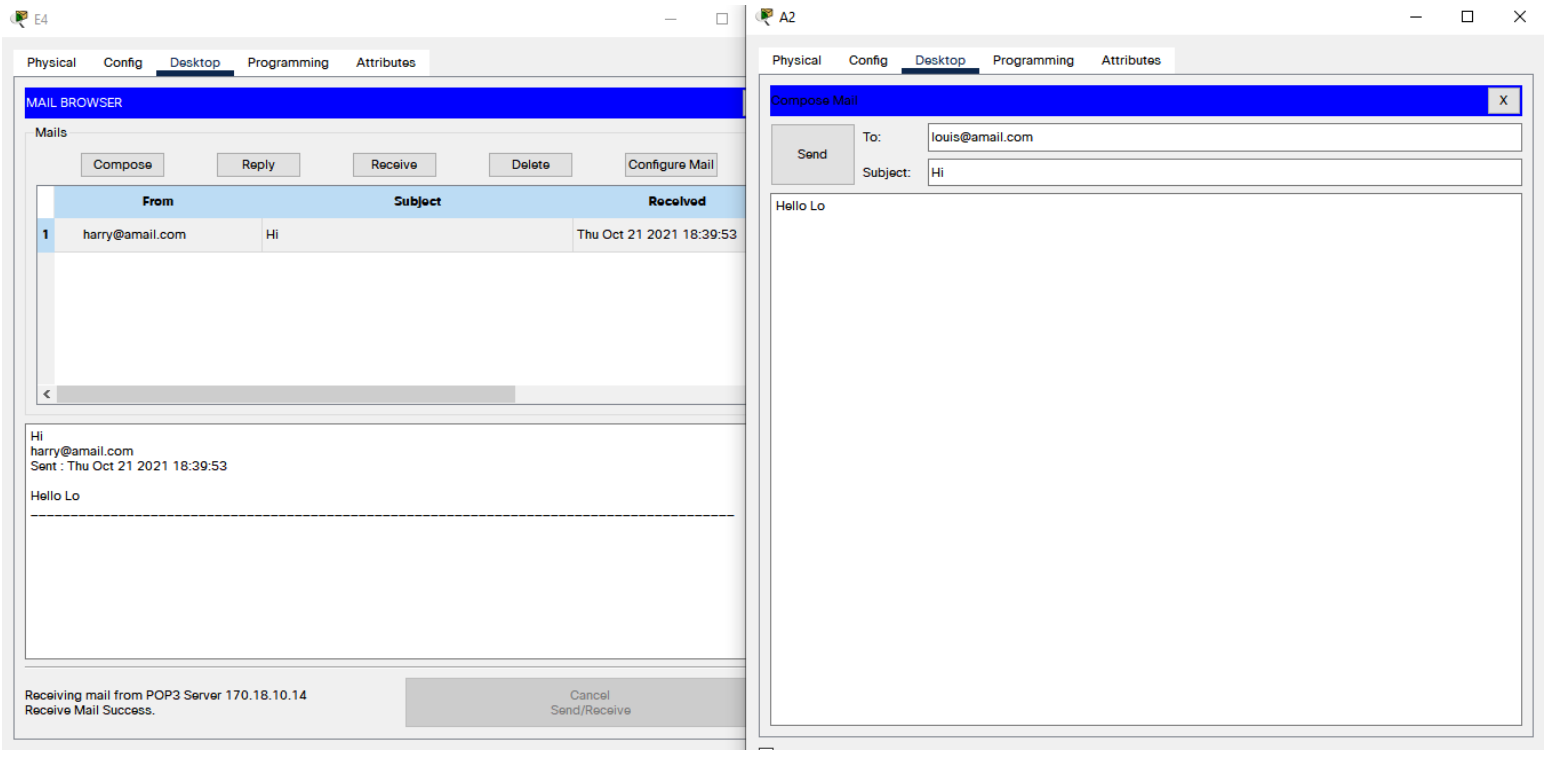


liam@bmail.com
greg@bmail.com

>> liam@bmail.com to harry@amail.com



>> harry@amail.com to louis@amail.com



CLOUD CONCEPTS:

Cloud computing is the **on-demand** delivery of storage, applications, database and other IT resources via the internet with **pay as you go** pricing

Hardware -> Software

- Type 1 Hypervisor - runs directly on the host machine's physical hardware
- Type 2 Hypervisor - installed on top of an existing OS

GobackN and Selective Repeat

Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in peer-to-peer mode and demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer-to-peer mode.

CODE:

Server.java

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;
import java.net.SocketException;

public class Server {
    static ServerSocket Serversocket;
    static DataInputStream dis;
    static DataOutputStream dos;

    public static void main(String[] args) throws SocketException {

    try {
        int a[] = { 30, 40, 50, 60, 70, 80, 90, 100, 110 };
        Serversocket = new ServerSocket(8011);
        System.out.println("waiting for connection");
        Socket client = Serversocket.accept();
        dis = new DataInputStream(client.getInputStream());
        dos = new DataOutputStream(client.getOutputStream());
        System.out.println("The number of packets sent is:" +
a.length);
        int y = a.length;
        dos.write(y);
        dos.flush();

        for (int i = 0; i < a.length; i++) {
            dos.write(a[i]);
            dos.flush();
        }

        int k = dis.read();

        dos.write(a[k]);
        dos.flush();

        } catch (IOException e) {
            System.out.println(e);
        } finally {
            try {
                dis.close();
```

```
                dos.close();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

        }
    }
}
```

Client.java

```
import java.lang.System;
import java.net.*;
import java.io.*;

public class Client {
    static Socket connection;

    public static void main(String a[]) throws SocketException {
        try {
            int v[] = new int[9];
            //int g[] = new int[8];
            int n = 0;
            InetAddress addr = InetAddress.getByName("Localhost");
            System.out.println(addr);
            connection = new Socket(addr, 8011);
            DataOutputStream out = new DataOutputStream(
                    connection.getOutputStream());
            DataInputStream in = new DataInputStream(
                    connection.getInputStream());
            int p = in.read();
            System.out.println("No of frame is:" + p);

            for (int i = 0; i < p; i++) {
                v[i] = in.read();
                System.out.println(v[i]);
                //g[i] = v[i];
            }
            v[5] = -1;
            for (int i = 0; i < p; i++)
             {
                System.out.println("Received frame is: " + v[i]);

                }
            for (int i = 0; i < p; i++)
                if (v[i] == -1) {
            System.out.println("Request to retransmit packet no "
                            + (i+1) + " again!!");
                    n = i;
                    out.write(n);
                    out.flush();
                }
```

```
                System.out.println();

                    v[n] = in.read();
                System.out.println("Received frame is: " + v[n]);



                System.out.println("quiting");
            } catch (Exception e) {
                System.out.println(e);
            }

        }
    }
```

OUTPUT:

Server.java

```
<terminated> Server [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe  (02-Nov-2021, 1:00:42 pm – 1:00:54 pm)
waiting for connection
The number of packets sent is:9
```

Client.java

```
<terminated> Client [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe  (02-Nov-2021, 1:00:54 pm – 1:00:54 pm)
Localhost/127.0.0.1
No of frame is:9
30
40
50
60
70
80
90
100
110
Received frame is: 30
Received frame is: 40
Received frame is: 50
Received frame is: 60
Received frame is: 70
Received frame is: -1
Received frame is: 90
Received frame is: 100
Received frame is: 110
Request to retransmit packet no 6 again!!

Received frame is: 80
quiting
```