**19CSE301-Computer Networks**

1. Place your department level details with subnetting scheme calculation
2. The following files has to be uploaded
3. Socket program-client ,server,text files/csv files
4. Cisco packet tracer – Application layer protocols
5. Cisco packet tracer – Virtual Local Area Network
6. Cisco packet tracer – OSPF
7. Cisco packet tracer – RIP
8. GoBack N and Selective Repeat protocol
9. Cloud concepts
10. **Make sure all Network demos should have the cisco packet tracer file attached**
11. All screenshots for cisco packet tracer in word document should show within LAN and across LAN

**Title : District Suraksha**

**List of Departments in the case study with the purpose**

|  |  |  |
| --- | --- | --- |
| **Department Name** | **Purpose** | **Network Details**  **(No of nodes,servers,Protocols)** |
| **RECEIVING DEPARTMENT** | **It used to receive the complaints from the citizens in a district.** | **No of nodes = 18**  **No of servers = 5(ftp dns smtp server)**  **No of protocols = 5** |
| **VERIFICATION DEPARTMENT** | **It is used to verify the details of the complaints.** | **No of nodes = 16**  **No of servers = 4(ftp dns smtp server)**  **No of protocols = 5** |
| **EXECUTION DEPARTMENT** | **It is used to execute the complaints which is verified by the verification department.** | **No of nodes = 18**  **No of servers = 2(ftp dns smtp server)**  **No of protocols = 5** |

|  |  |  |
| --- | --- | --- |
| **Group Member Roll No** | **Name** | **Department** |
| **CB.EN.U4CSE19105** | **A.V.V.L.N BALARAM** | **RECEIVING DEPARTMENT** |
| **CB.EN.U4CSE19137** | **P.S.V.AKASH** | **VERIFICATION DEPARTMENT** |
| **CB.EN.U4CSE19154** | **T. SAI JAYANTH** | **EXECUTION DEPARTMENT** |

**Case study**

|  |  |  |
| --- | --- | --- |
| **Topic** | **Page No in the document** | **Supporting file name** |
| Problem Statement | **3** |  |
| Objective of the case study | **3** |  |
| Network Architecture Diagram | **3** |  |
| Performance Parameters | **3-5** |  |
| Department Details in the case study with description | **6** |  |
| Subnet ip scheme followed  [Solving the IP address allocation process for each subnetwork ] | **6-12** |  |
| Socket programming | **13-17** | **CLIENT1.py**  **CLIENT2.py**  **SE\_TEST.py**  **problem.csv** |
| Cisco packet tracer – Application layer protocols | **18-24** | **CN CASE STUDY** |
| Cisco packet tracer – Virtual Local Area Network | **27-28** | **CN CASE STUDY** |
| Cisco packet tracer – OSPF | **25** | **CN CASE STUDY** |
| Cisco packet tracer – RIP | **26** | **CN CASE STUDY** |
| GoBack N and Selective Repeat protocol | **29-41** | **Server\_.java**  **Client\_.java**  **Client\_selective.java**  **Server\_selective.java** |
| Cloud concepts | **40-42** |  |
| Cloud and Cisco packet tracer |  |  |
| Cloud and Networking |  |  |
| How is cloud related to your application? | **43** |  |

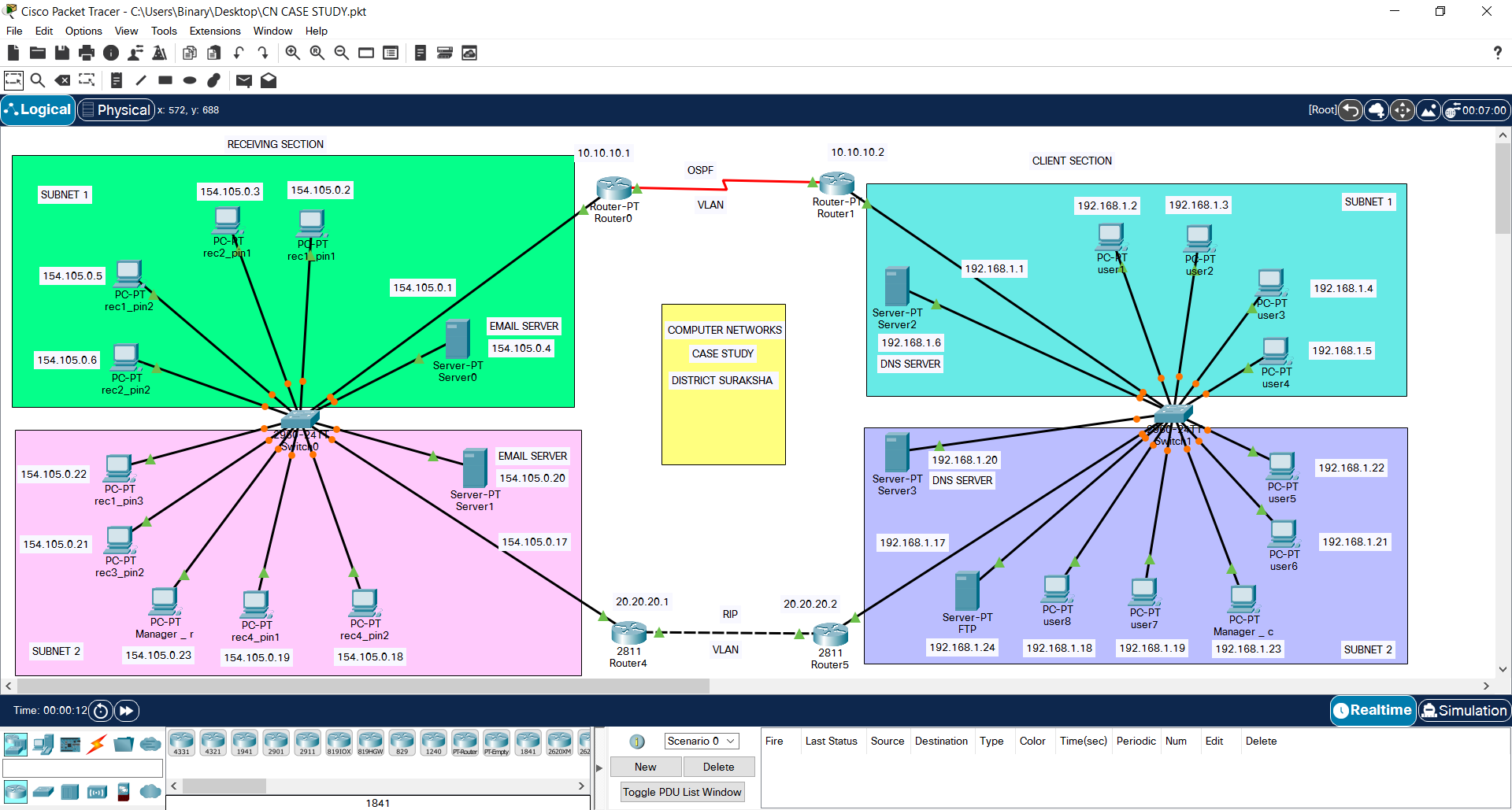
**Problem Statement:**

In olden days most of the people write complaints in the form of letters and post it to the certain officials about their problem and the issues that are facing by them due to that. As the time grows, we need to develop our standards So, why don’t we transform that into digital.

**Objective of the Case Study:**

Our objective is to resolve the problems by taking the complaints digitally and solving it done by the government as early as possible with an verified details of each and every complaint.

**Network Architecture Diagram:**



**List of Network performance parameters:**

Normally the performance of a network is used to measure the service quality of a network as perceived by the user. To measure the performance of a network there are different ways depending on design of the network.

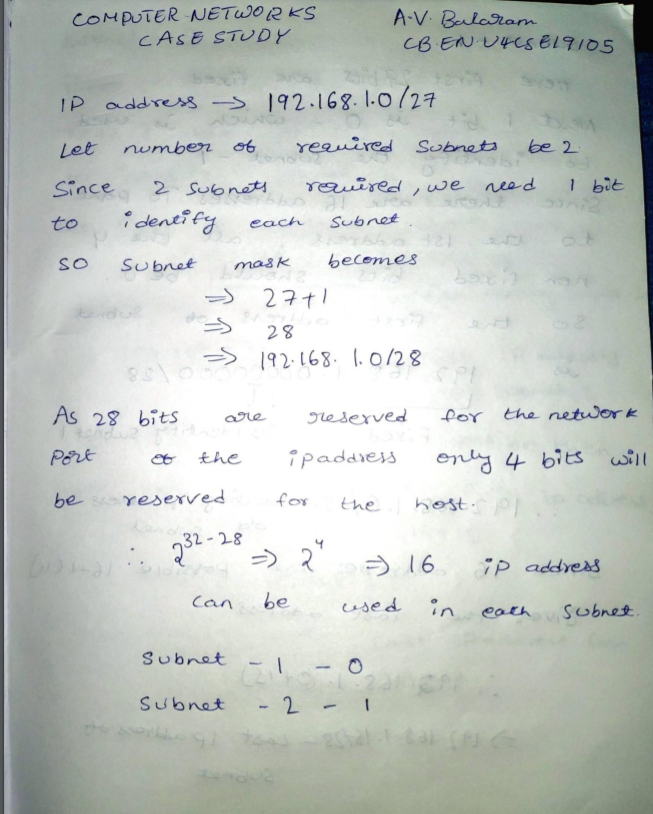
|  |  |  |
| --- | --- | --- |
| Parameters | Definition | Formula |
| Bandwidth | The maximum amount of data transmitted over an internet connection in a given amount of time. It determines how rapidly the web server is able to upload the requested information.  Digital Devices: bps(bytes per second)  Analog Devices: cps(cycles per second)  It is a potential measurement  of a link. | Expressed as bits per second (bps), modern network links have greater capacity, which is typically measured in millions of bits per second (megabits per second, or Mbps) or billions of bits per second (gigabits per second, or Gbps). |
| Throughput | The number of messages successfully transmitted per unit time. The maximum throughput of a network may be consequently higher than the actual throughput achieved in every day consumption.  It is an actual measurement of how fast we can send the data. | R = I/T  R : Rate(Throughput) I : Inventory  T : Time |
| Transmission time | The time required for transmission of a message depends on the size of the message and the bandwidth of the channel. | Transmission time=Message size / Bandwidth |
| Propagation Time | Propagation time measures the time required for a bit to travel from the source to the destination. The propagation time is calculated by dividing the distance by the propagation speed. | Propagation time = Distance  /Propagation speed |
| Processing Delay | Time taken by the processor to process the data packet is called processing delay. |  |

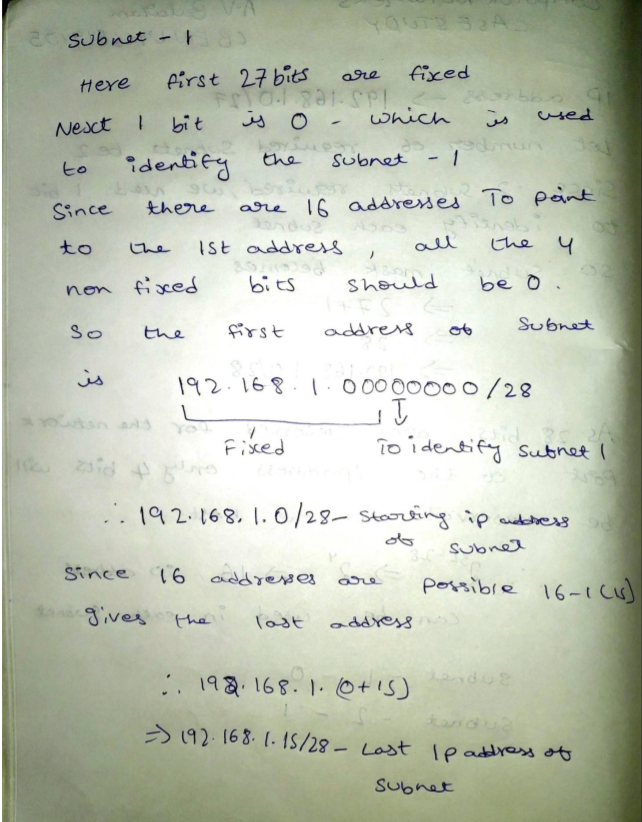
|  |  |  |
| --- | --- | --- |
| Queuing Delay | Time spent by the data packet waiting in the queue before it is taken for execution is called queuing delay. |  |
| Packet Loss | Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination. Due to network congestion | Efficiency = 100% \* (transferred - retransmitted) / transferred  Network Loss = 100 - Efficiency |
| Latency | The time required to successfully send a packet across the network.  The total time taken for a complete message to arrive at the destination, starting with the time when the first bit of the message is sent out from the source and ending with the time when the last bit of the message is delivered at the destination. Here Latency is also known as ping rate and  measured in milliseconds. | Latency = Propagation Time + Transmission Time + Queuing Time + Processing Delay.  Propagation Time = Distance / Propagation Speed |
| Jitter | Jitter is nothing but Packet delay Variance. The variation in the delay of received packets. It is considered as a problem when different packets of data face different delays in a network and the data at the receiver application is time sensitive i.e., audio or video data. It is measured in MilliSeconds(ms) | Latency=sum of all delays  To measure Jitter, we take the difference between samples, and then divide by the number of samples. |

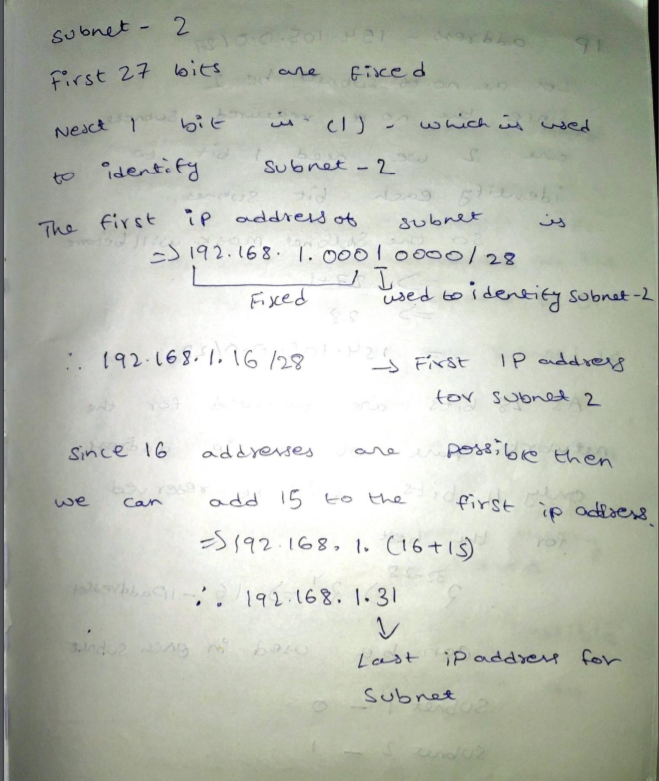
**Department Details in the case study with description:**

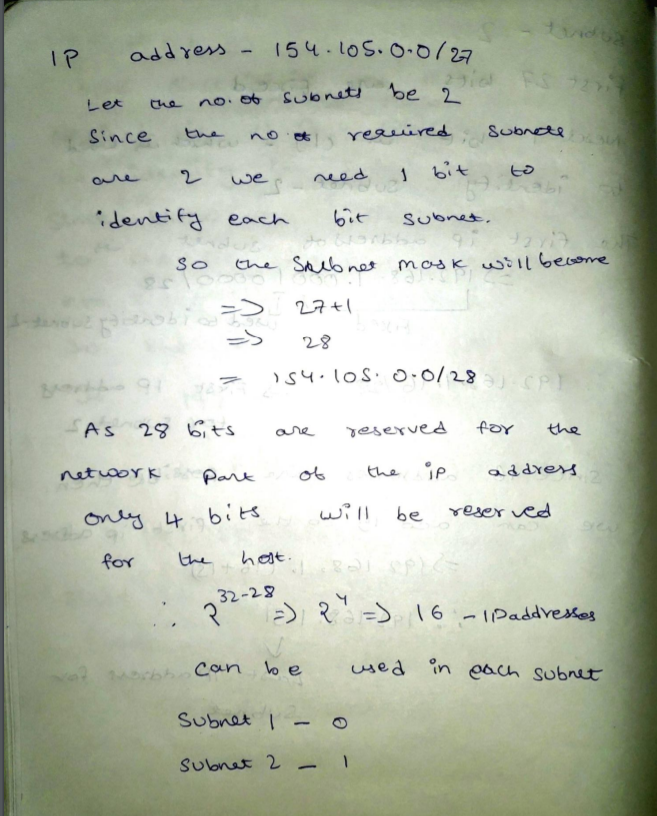
Here we use networks in order to communicate between multiple departments. Also, execution department will communicate with various other offices in order to execute the verified problems. So, we need networking here for Receiving, Verification, Execution.

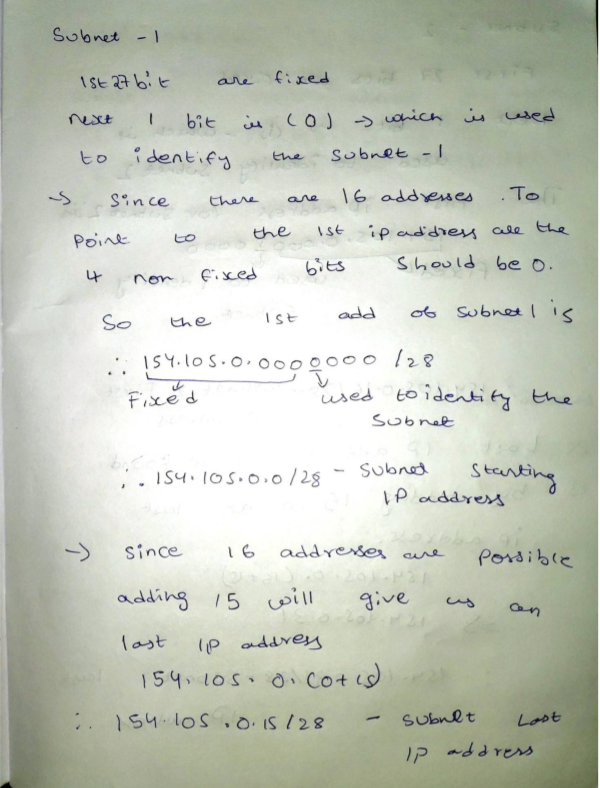
**Subnet IP Scheme:**

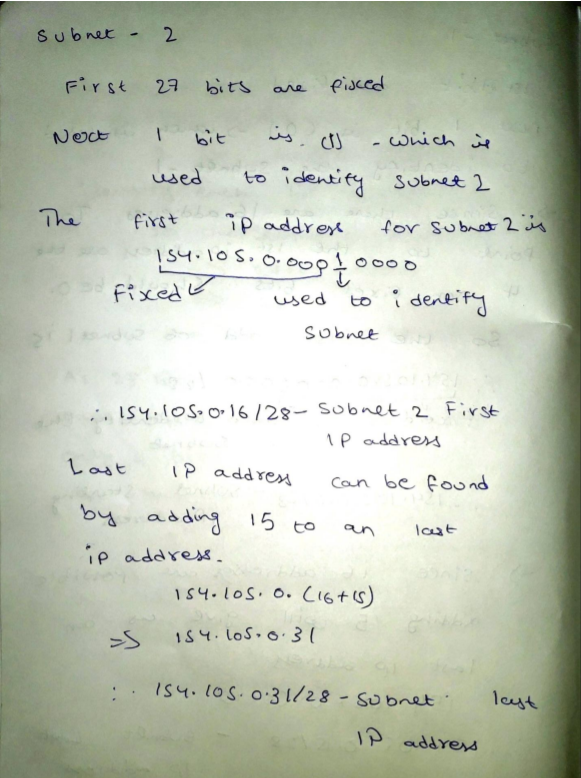


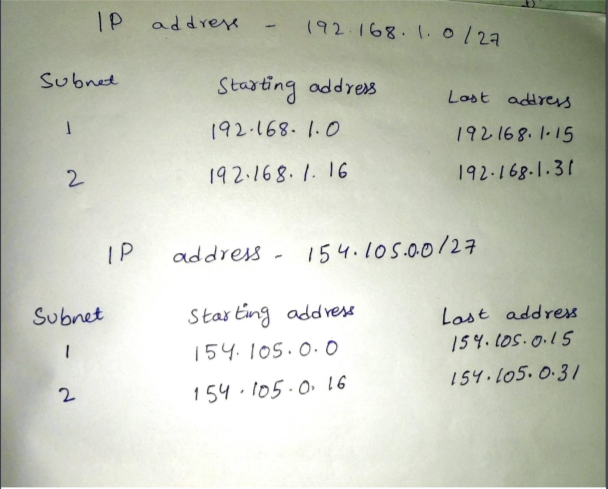












**Socket Programming:**

**File Handling Operations using Socket Programming**

1. **Description of the text file**

We are having 6 columns in an csv file. Which is very important to validate and whenever the code got run in cmd prompt it will be reflected in an excel sheet also.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Phone No** | **District** | **Pin code** | **Area** | **Problem** |

**b. List of operations completed with the File:**

VIEW INSERT

**c. Client-Side program with output for each operation**

import socket

import pandas as pd

import numpy as np

df = pd.read\_csv(r"problem.csv")

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

s.connect((socket.gethostname(), 1024))

Name = input(s.recv(1000).decode("utf-8"))

s.send(Name.encode("utf-8"))

Phone\_No = int(input(s.recv(1000).decode("utf-8")))

s.send(str(Phone\_No).encode("utf-8"))

District= str(input(s.recv(1000).decode("utf-8")))

s.send(District.encode("utf-8"))

Pincode = int(input(s.recv(1000).decode("utf-8")))

s.send(str(Pincode).encode("utf-8"))

Area = str(input(s.recv(1000).decode("utf-8")))

s.send(Area.encode("utf-8"))

Problem = str(input(s.recv(1000).decode("utf-8")))

s.send(Problem.encode("utf-8"))

print(Name)

print(Phone\_No)

print("Thanks for registering your problems....")

s.close()

**d. Server-Side program with output for each operation**

import socket

import pandas as pd

import numpy as np

from \_thread import \*

df = pd.read\_csv(r"problem.csv")

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

s.bind((socket.gethostname(), 1024))

ThreadCount = 0

s.listen(10)

list1=[]

def threaded\_client(est):

while True:

est.send(bytes("Enter Name : ", "utf-8"))

Name = est.recv(1000).decode("utf-8")

est.send(bytes("Enter Phone number:","utf-8"))

Phone\_No= est.recv(1000).decode("utf-8")

Phone\_No= int(Phone\_No)

est.send(bytes("Enter District : ", "utf-8"))

District = est.recv(1000).decode("utf-8")

est.send(bytes("Enter Pincode:","utf-8"))

Pincode = est.recv(1000).decode("utf-8")

Pincode= int(Pincode)

est.send(bytes("Enter Region : ", "utf-8"))

Area = est.recv(1000).decode("utf-8")

est.send(bytes("Enter Problem : ", "utf-8"))

Problem = est.recv(1000).decode("utf-8")

list1 = [[Name,Phone\_No,District,Pincode,Area,Problem]]

df2 = pd.DataFrame(list1, columns=['Name','Phone\_No','District','Pincode','Area','Problem'])

df2.to\_csv(r'problem.csv',mode='a',index=False,header=False)

est.close()

while True:

est, addr = s.accept()

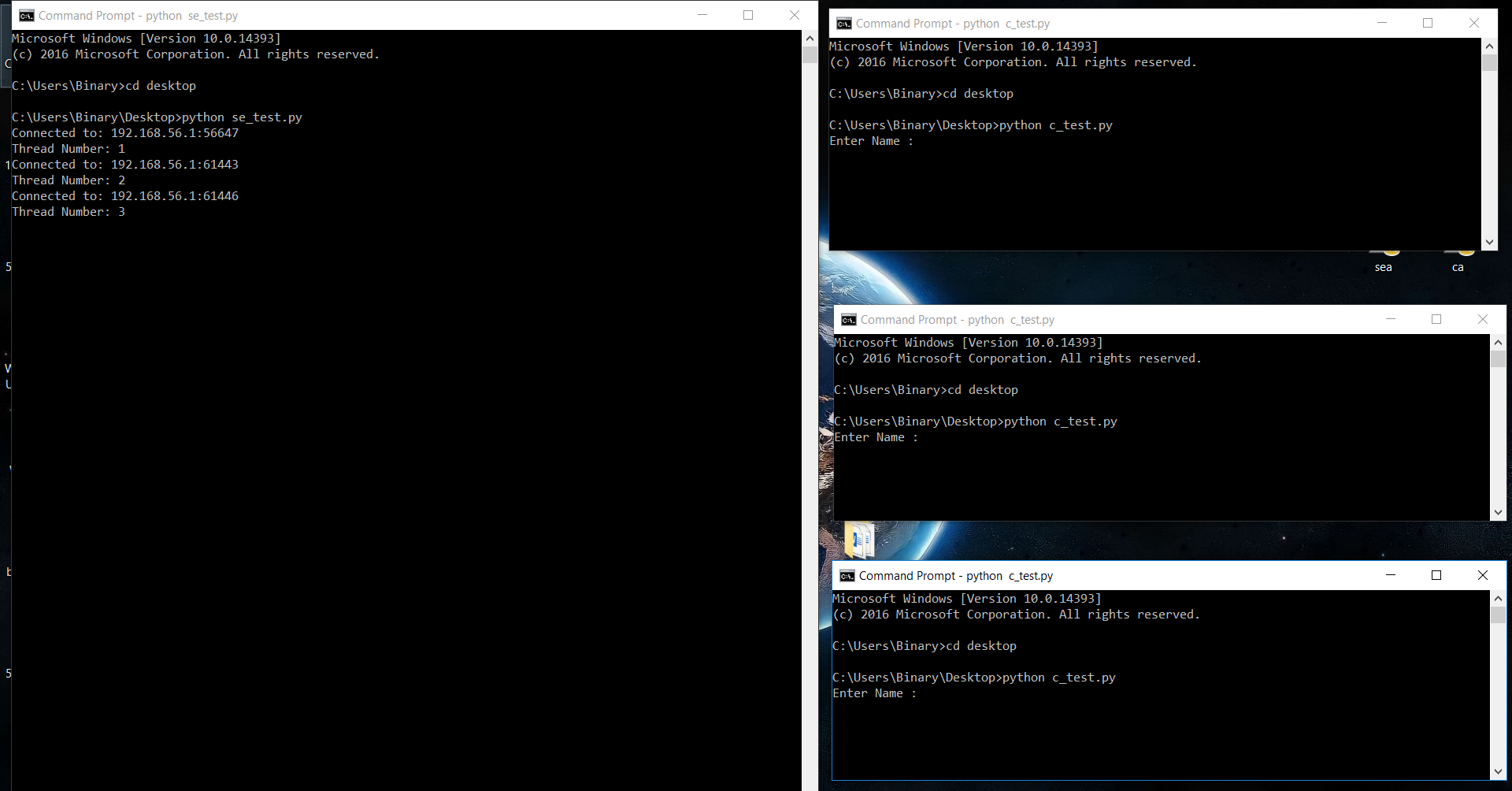
print('Connected to: ' + addr[0] + ':' + str(addr[1]))

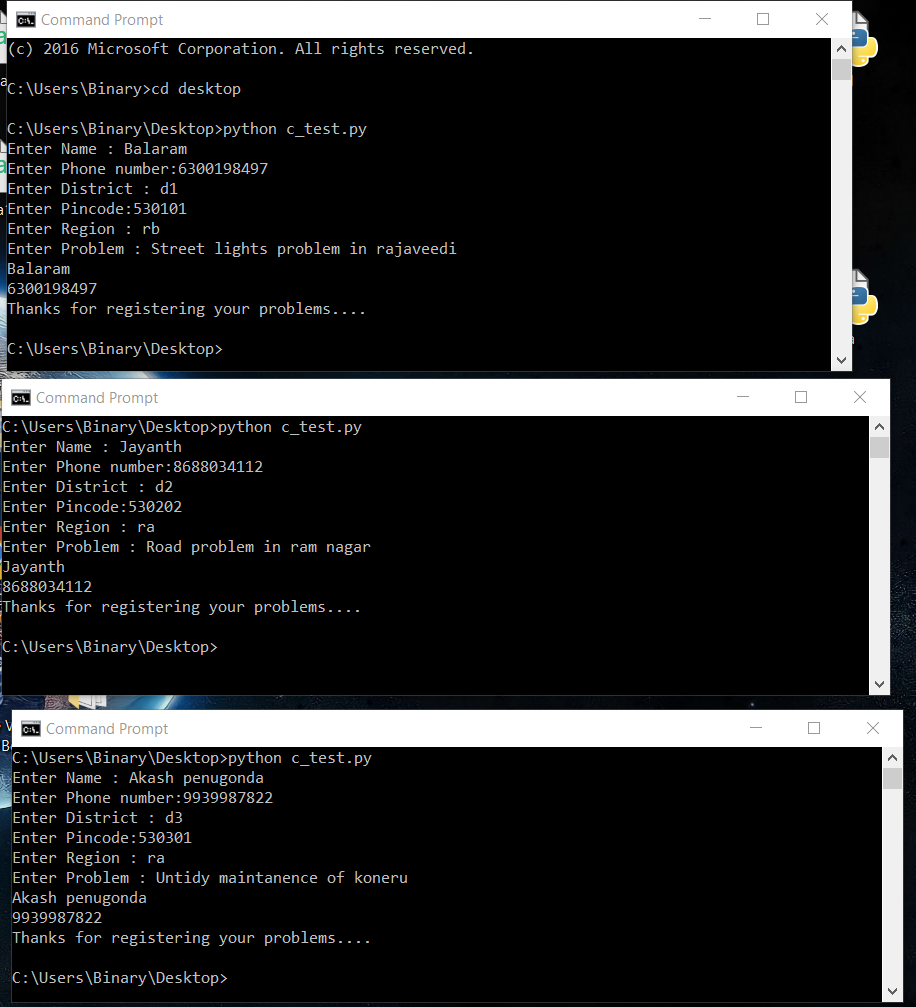
start\_new\_thread(threaded\_client, (est, ))

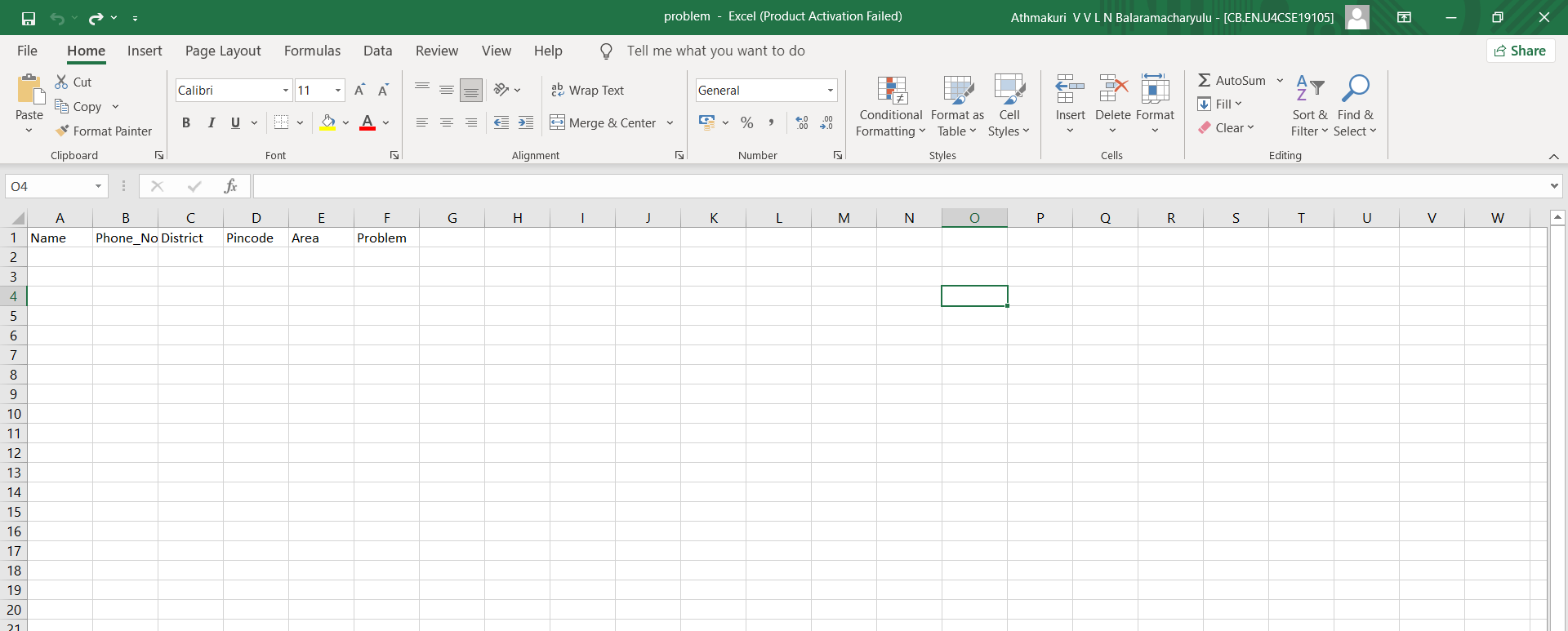
ThreadCount += 1

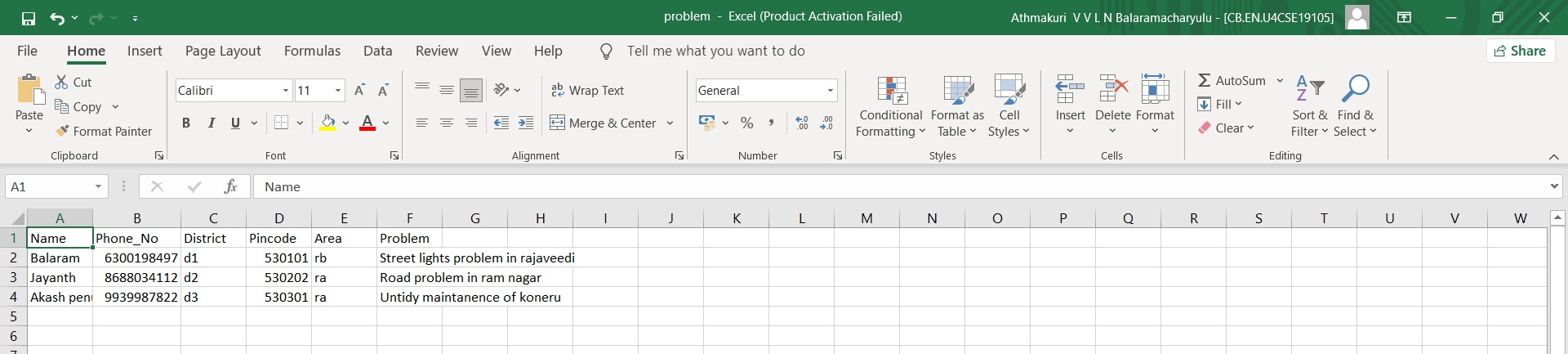
print('Thread Number: ' + str(ThreadCount))

**OUTPUT FOR BOTH CLIENTS AND SERVER**

**Insertion/view:**



**Excel sheet initially:**

**Output after input reflection in excel:**

**Cisco packet tracer – Application layer protocols**

MODEL:

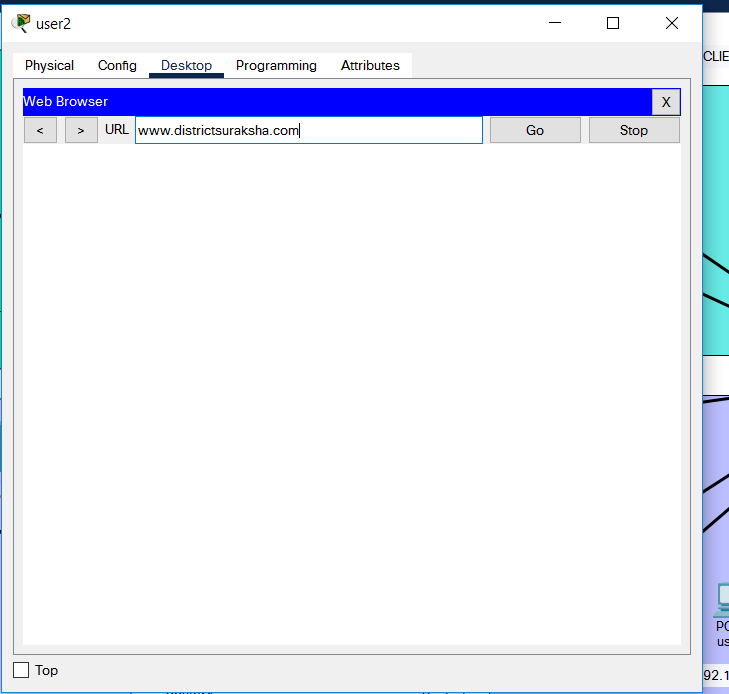
Here in each sections computers divided into 2 subnetworks and each subnetwork consists of 4 computers and the servers present in each subnet belongs to it.

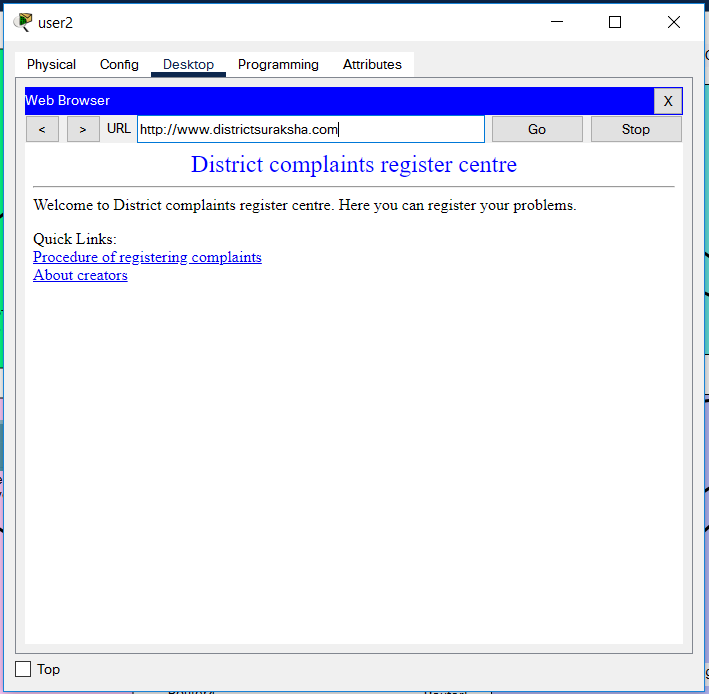
Here 10 users there in clients section. Here 10 employees there in receiving section. In each section 2 managers will be there.

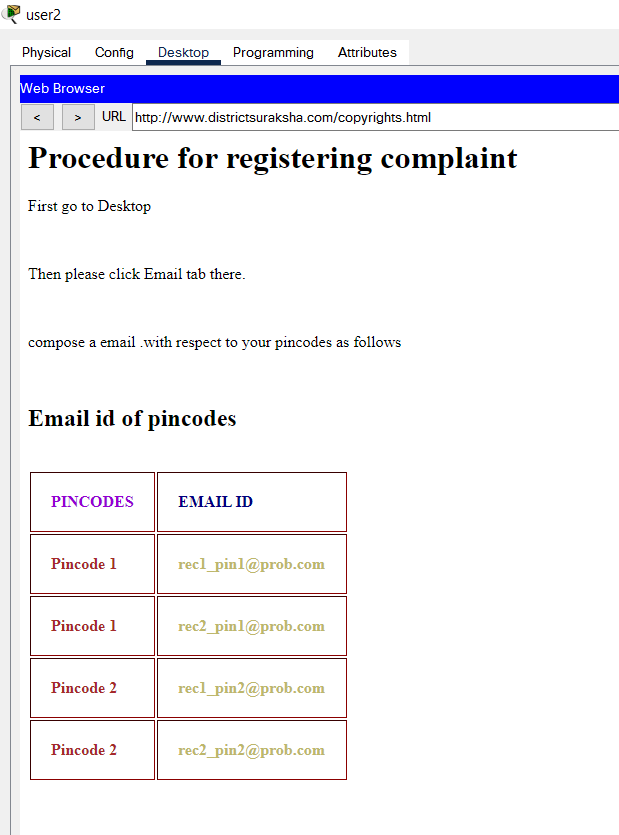
Communication will be there via email between clients (that is users) and employees in receiving section. File transfer takes between managers regarding daily reports.

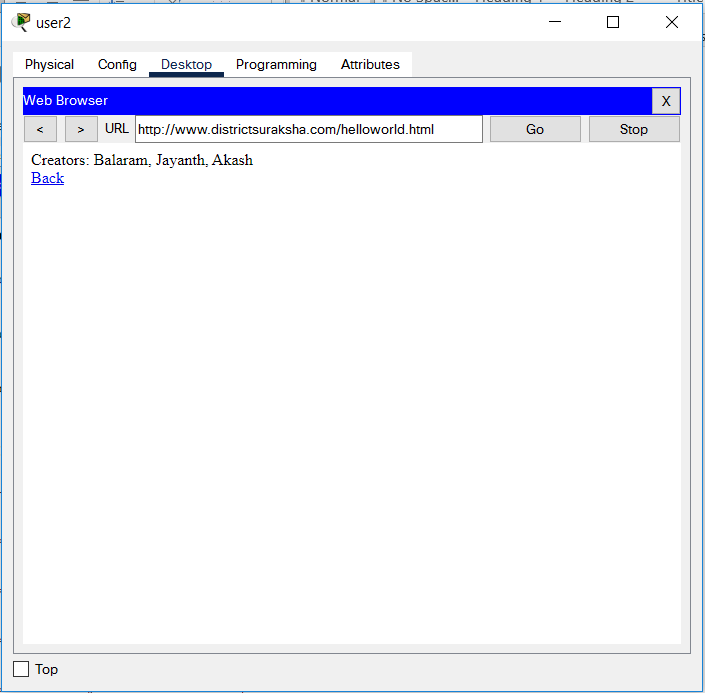
Let’s see implementation of each concept.

DNS:

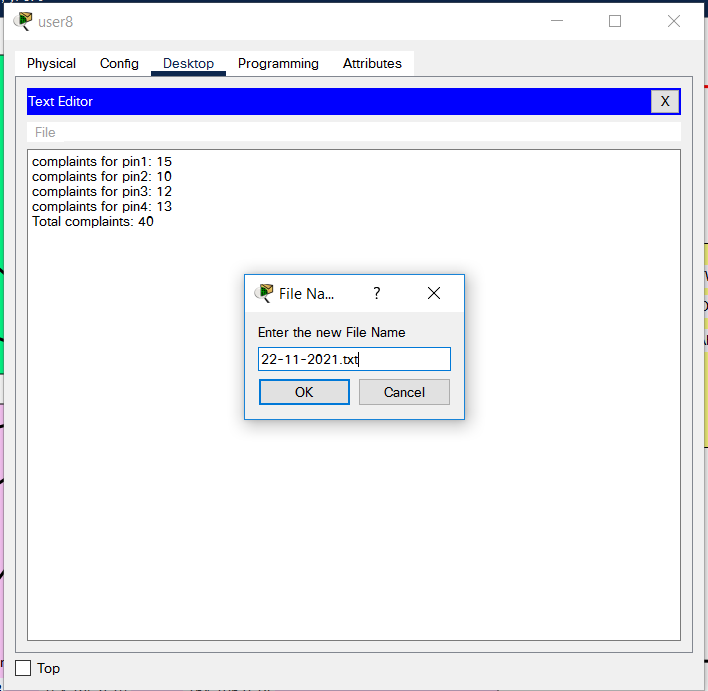




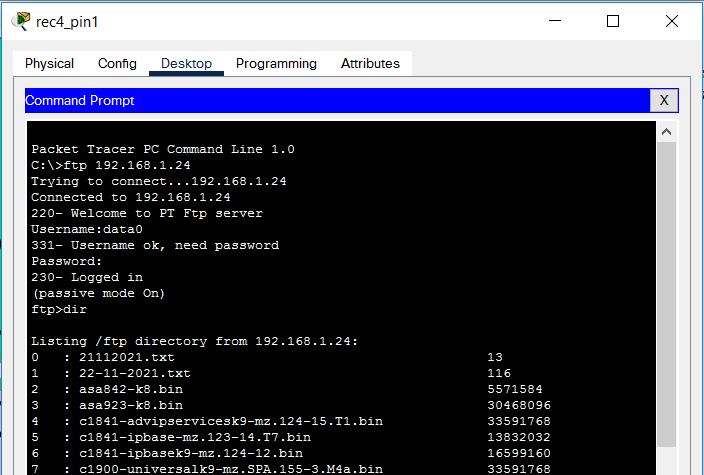


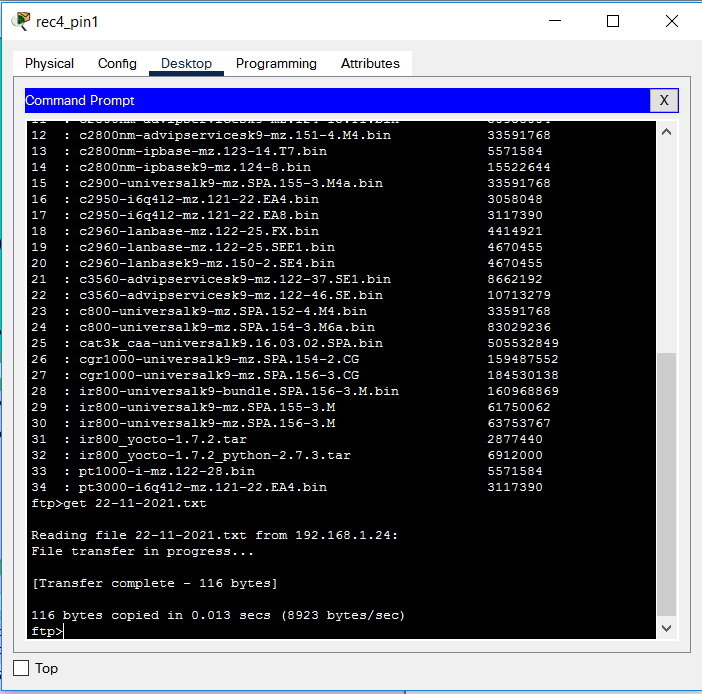


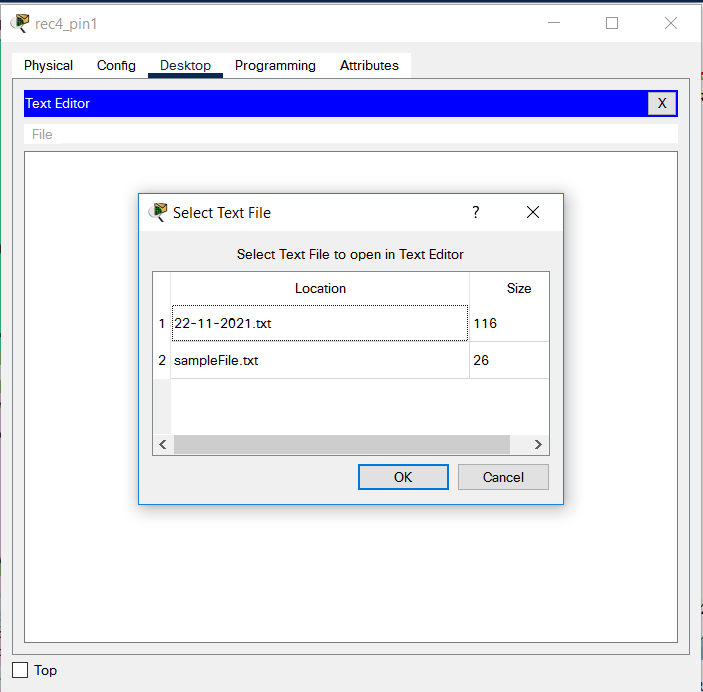
FTP SERVER:



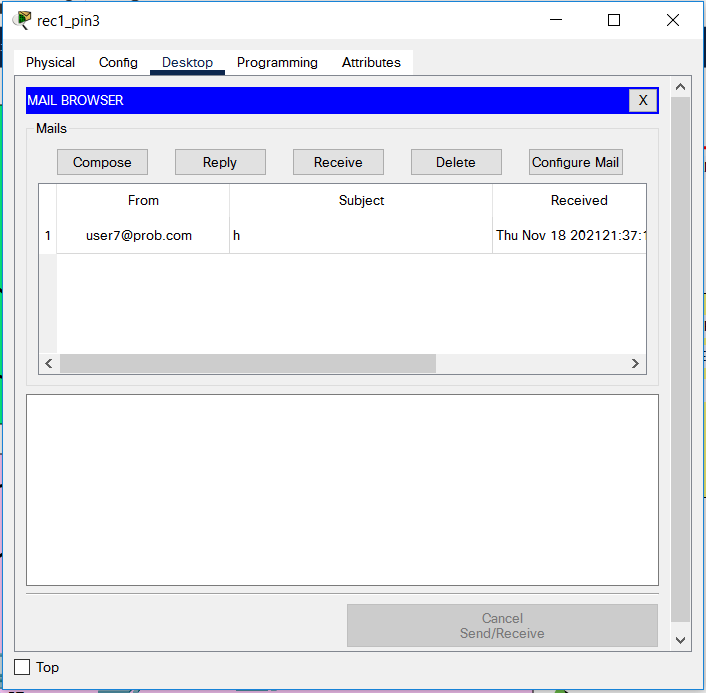




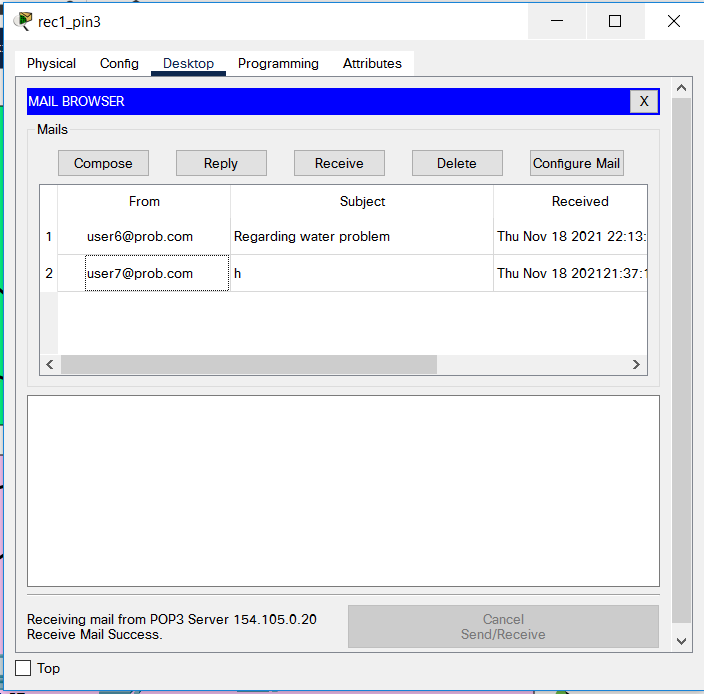


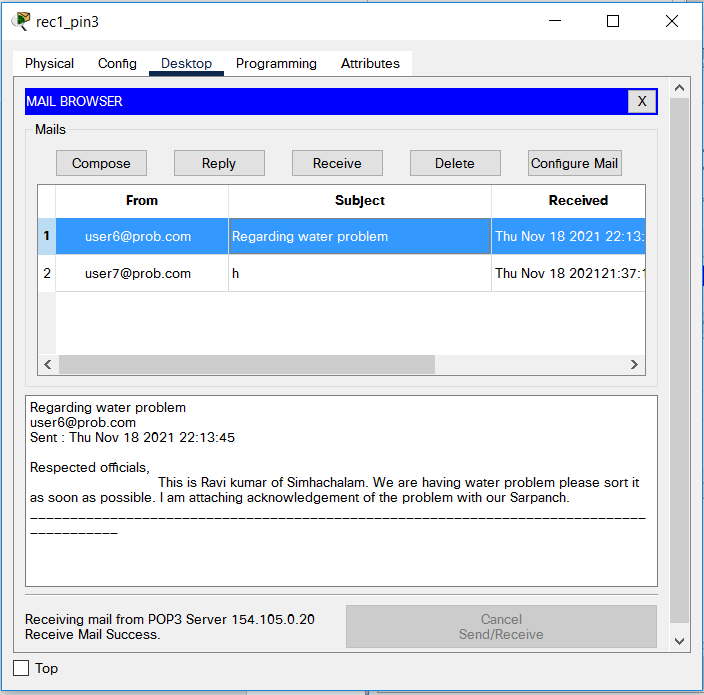


SMTP:

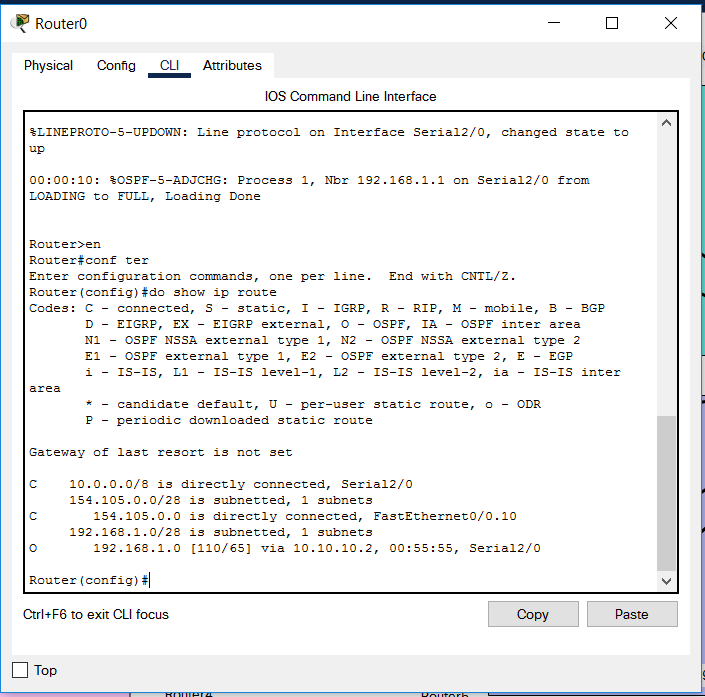


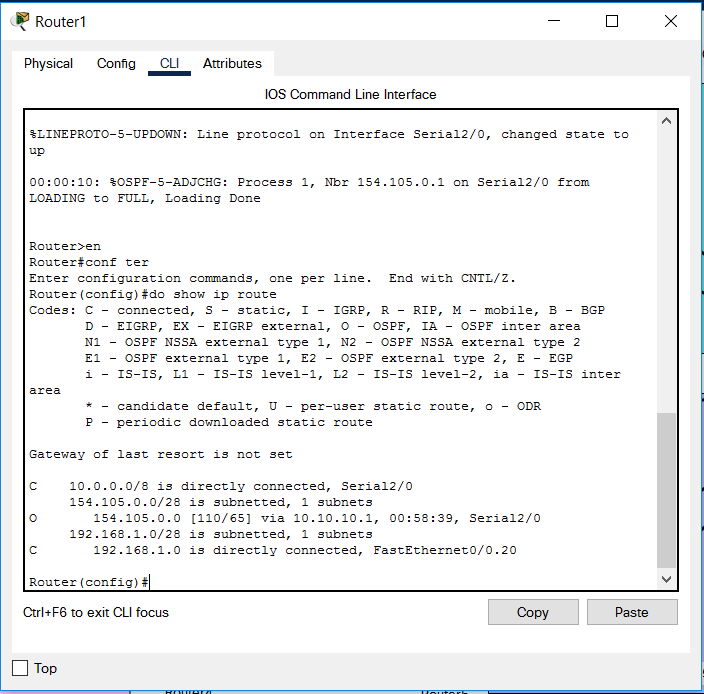




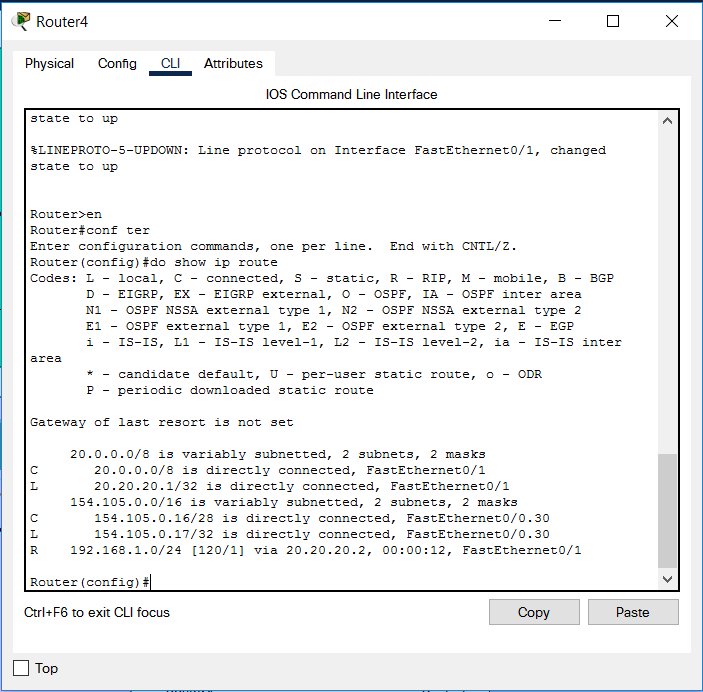


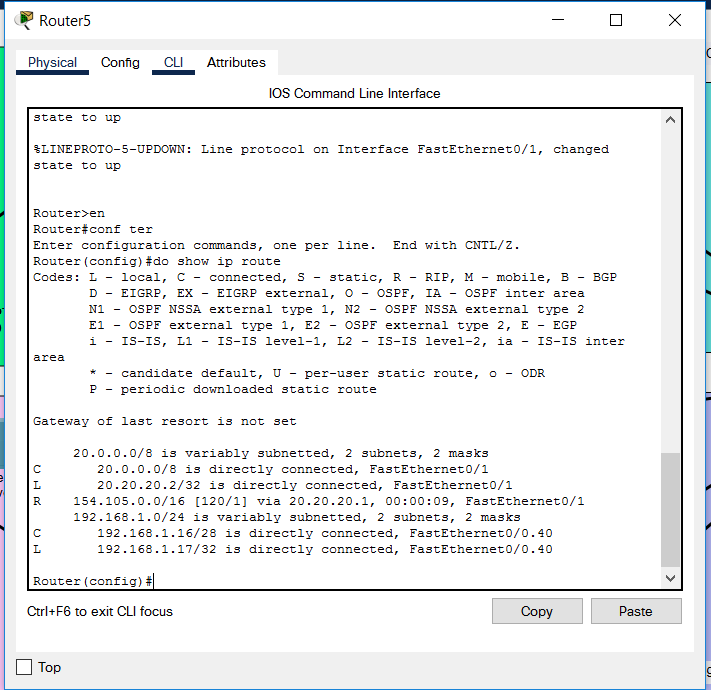
OSPF:



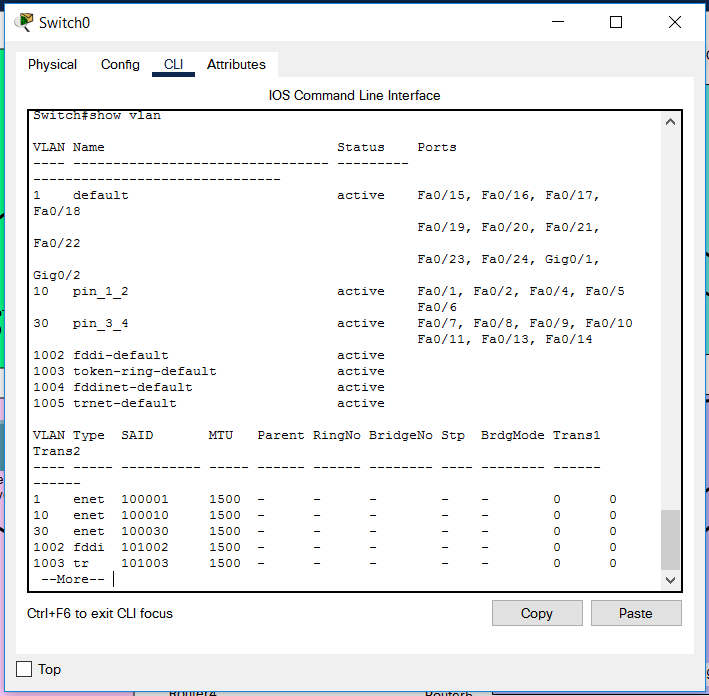


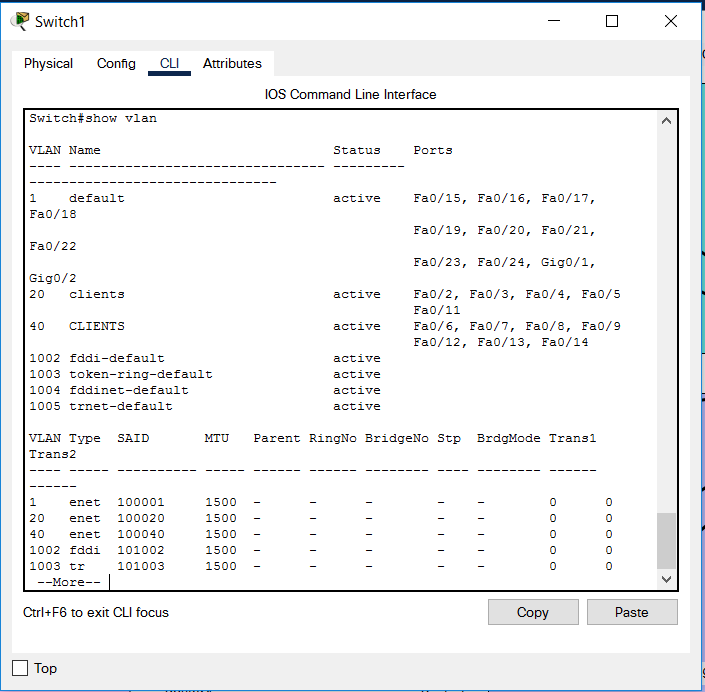
RIP:

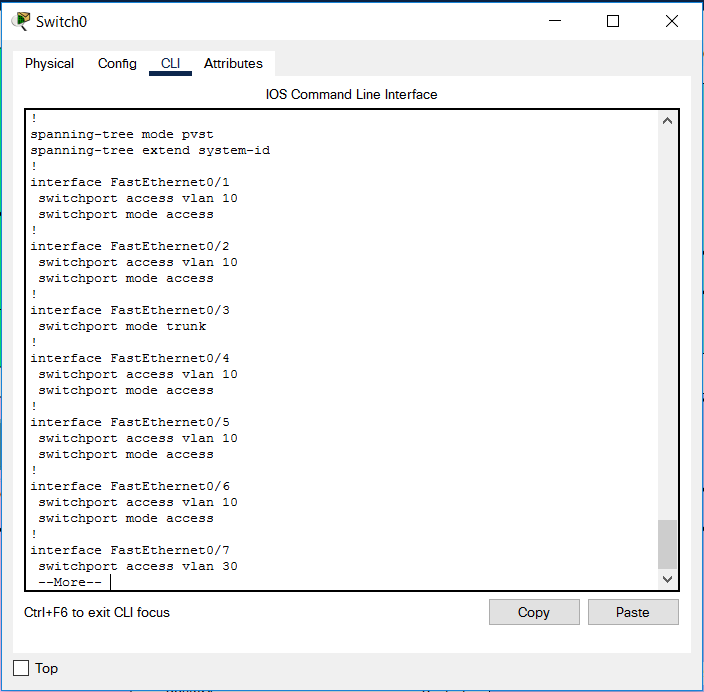


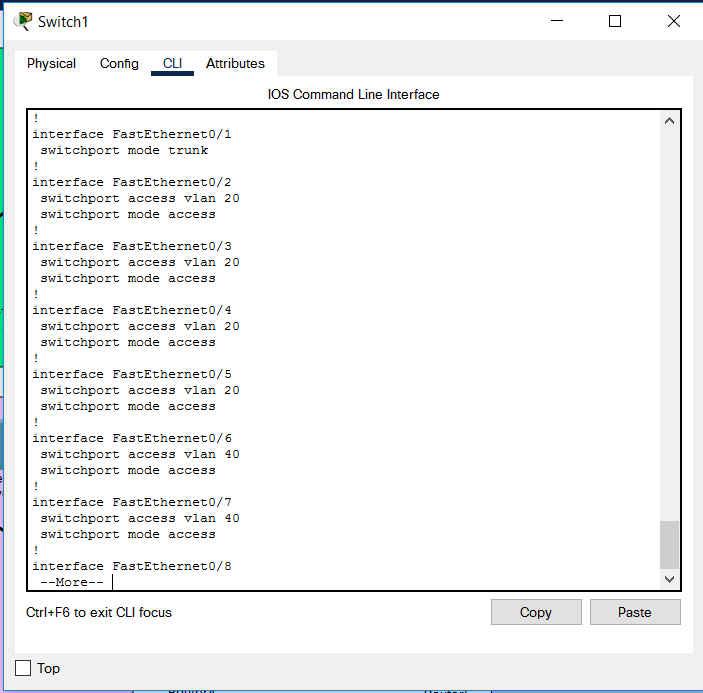


VLAN:









**GoBack N and Selective Repeat protocol**

**GoBack N PROTOCOL**

**server\_.java**

import java.io.\*;

import java.net.\*;

import java.util.\*;

class server\_{

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

System.out.println("...Server...");

System.out.println("...Waiting...");

InetAddress address = InetAddress.getByName("Localhost");

ServerSocket ss = new ServerSocket(500);

Socket s1 = new Socket();

s1 = ss.accept();

BufferedInputStream in = new BufferedInputStream(s1.getInputStream());

DataOutputStream out = new DataOutputStream(s1.getOutputStream());

System.out.println("received request for sending frames");

int n = in.read();

boolean[] array = new boolean[n];

int pc = in.read();

System.out.println("...Sending...");

if(pc==0){

for(int i=0;i<n;i++){

System.out.println("Sending frame => "+i);

out.write(i);

out.flush();

System.out.println("..Waiting for acknowledge..");

try{

Thread.sleep(5000);

}

catch (Exception e){}

int a = in.read();

System.out.println("received acknowledgment for frame => " +i+ " as "+a);

}

out.flush();

}

else{

for(int i=0;i<n;i++){

if(i==3) {

System.out.println("Sending frame number => " +i);

}

else{

System.out.println("sending frame no => " +i);

out.write(i);

out.flush();

System.out.println("Waiting for acknologment ");

try {

Thread.sleep(7000);

}

catch(Exception e){}

int a = in.read();

if(a!=255){

System.out.println("received ack for frame num =>"+i+" as "+a);

array[i]=true;

}

}

}

for(int a=0;a<n;a++){

if(array[a]==false){

System.out.println("Resending frame => " +a);

out.write(a);

out.flush();

System.out.println("waiting for ack ");

try {

Thread.sleep(5000);

}

catch(Exception e){}

int b = in.read();

System.out.println("receiving ack for frame num => "+a+" as "+b);

array[a]=true;

}

}

out.flush();

}

in.close();

System.out.println("Quiting");

}

}

**client\_.java**

import java.io.\*;

import java.net.\*;

import java.math.\*;

import java.util.\*;

class client\_{

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

InetAddress address = InetAddress.getByName("Localhost");

System.out.println(address);

Socket s1 = new Socket(address,500);

BufferedInputStream in = new BufferedInputStream(s1.getInputStream());

DataOutputStream out = new DataOutputStream(s1.getOutputStream());

Scanner sc = new Scanner(System.in);

System.out.println("...client...");

System.out.println("Connect");

System.out.println("Enter the num of frames to be request to server");

int c = sc.nextInt();

out.write(c);

out.flush();

System.out.println("Enter type of trans. Error =1 : No Error=0");

int choice = sc.nextInt();

out.write(choice);

int i=0,j=0,check =0;

if(choice==0){

for(j=0;j<c;j++){

i = in.read();

System.out.println("receiver frame number => " +i);

System.out.println("Sending acknowlwdgement for frame number=> "+i);

out.write(i);

out.flush();

}

out.flush();

}

else{

for(j=0;j<c;j++){

i = in.read();

if(i==check){

System.out.println("i => " +i+ "check => " +check);

System.out.println("received frame number => "+i);

System.out.println("sending acknowledgement for frame num => " +i);

out.write(i);

check++;

}

else{

j--;

System.out.println("Discarded frame no => " +i);

System.out.println("Sending negative ack ");

out.write(-1);

}

out.flush();

}

}

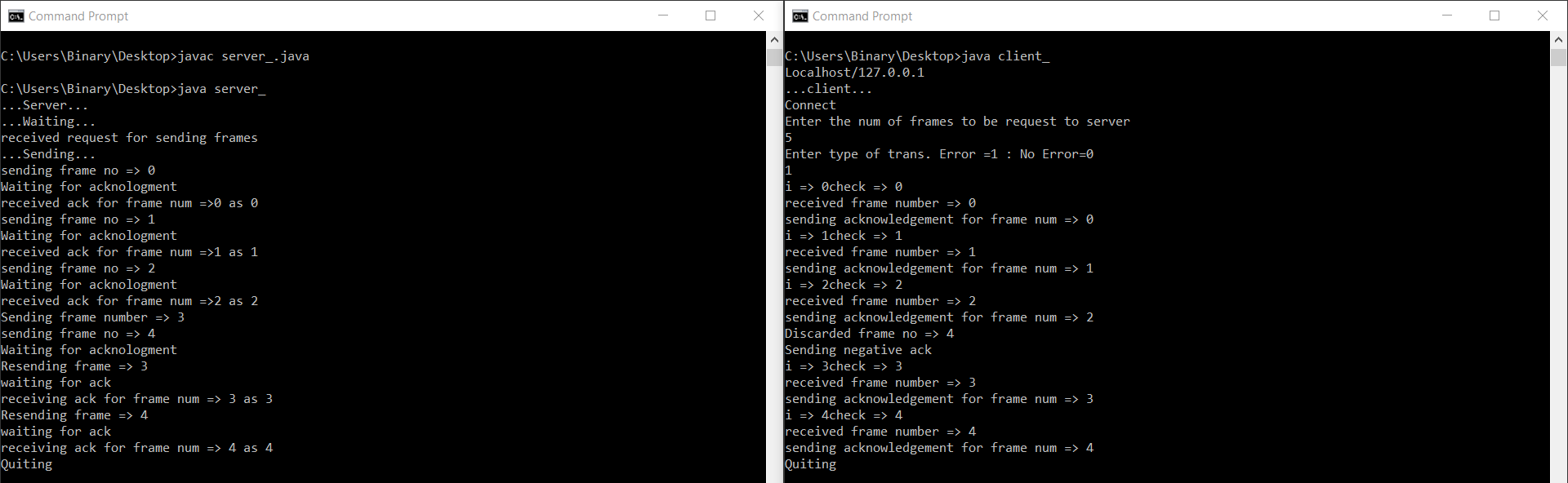
in.close();

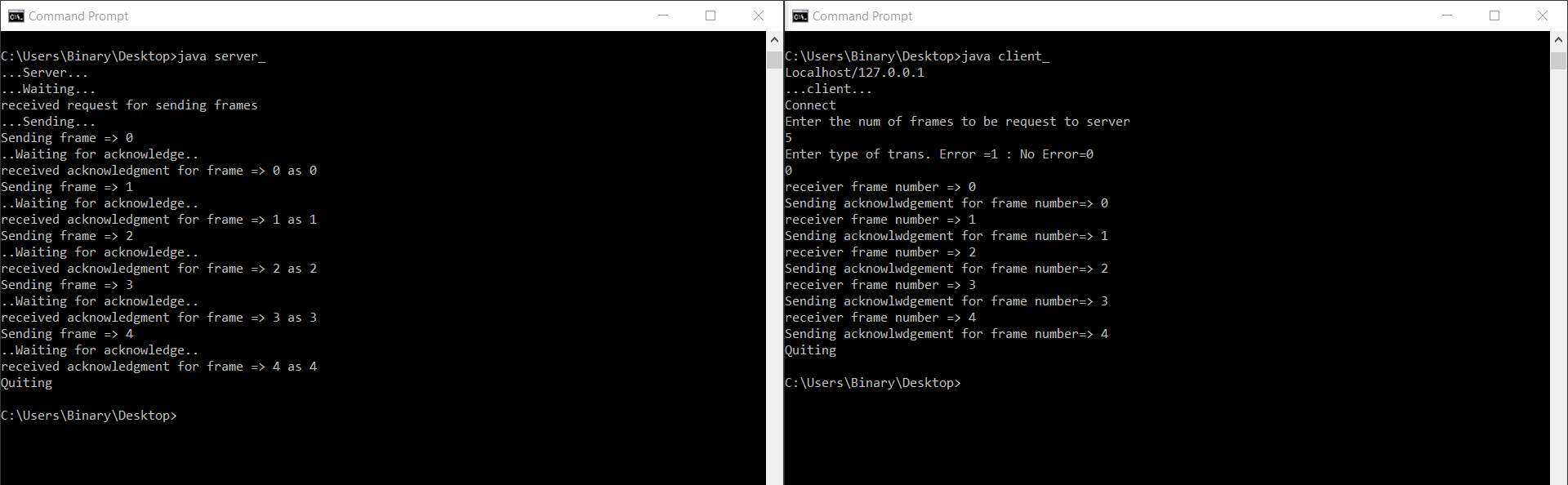
out.close();

System.out.println("Quiting");

}

}





**SELECTIVE REPEAT PROTOCOL**

**server\_selective.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;

class server\_selective

{

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 };

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)

{

e.printStackTrace();

}

}

}

}

**client\_selective.java**

import java.lang.System;

import java.net.\*;

import java.io.\*;

import java.text.\*;

import java.util.Random;

import java.util.\*;

class client\_selective {

static Socket connection;

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

try {

int v[] = new int[10];

int n = 0;

Random rands = new Random();

int rand = 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];

}

rand = rands.nextInt(p);

v[rand] = -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 FROM 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("QUITTING");

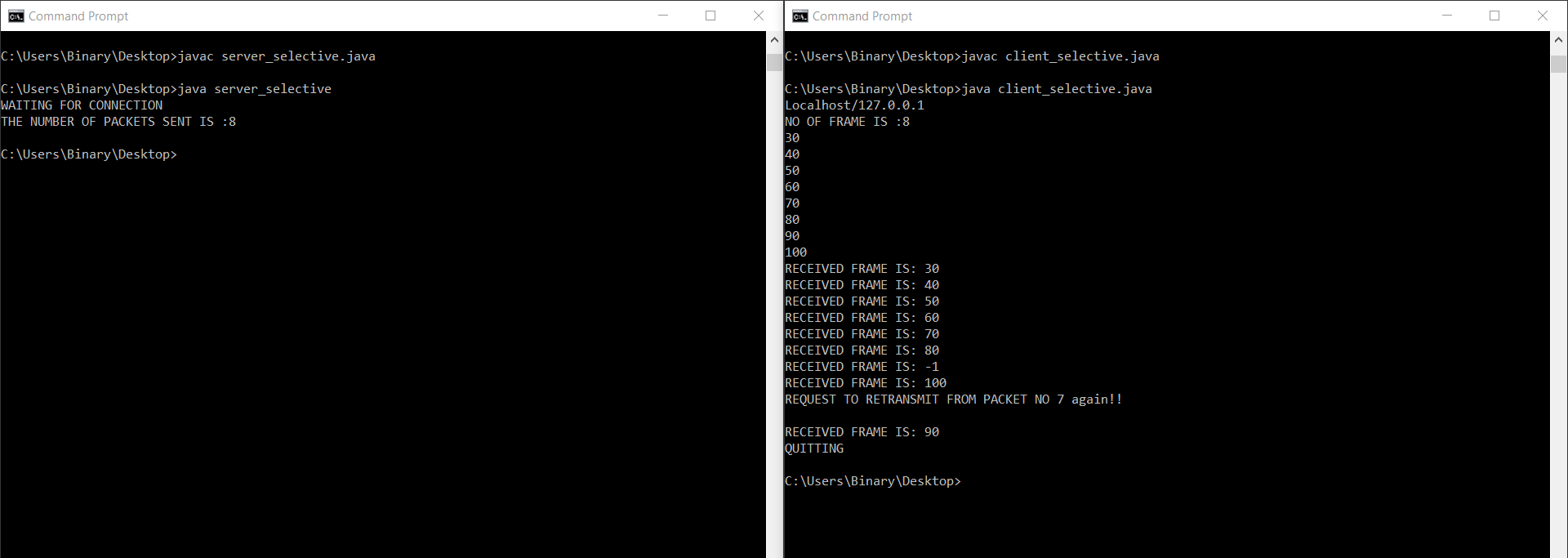
} catch (Exception e) {

System.out.println(e);

}

}

}



**Cloud concepts:**

cloud computing is the delivery of computing services—including servers, storage, databases, networking, software, analytics, and intelligence—over the Internet to offer faster innovation.

**Cloud networking is a sort of IT infrastructure in which a company's network capabilities and resources are hosted in a public or private cloud platform, or by a service provider, and available on demand. Companies can establish a private cloud network using on-premises cloud networking resources, or use public cloud networking services, or a hybrid cloud mix of the two. Virtual routers, firewalls, bandwidth and network management software are just a few of the network resources available, with additional tools and functions available as needed.** Cloud refers to software and services that run on the internet, instead of locally on your computer now a day’s cloud is very useful because there is a vast amount of data in our day-to-day life. Not only that now a day’s keeping personal servers are quite hard in terms of cost. Due to that we are moving to use the cloud as an infrastructure service. Virtualization in cloud computing allows a provider to virtualize servers, storage, or other physical hardware or data centre resources, which can then, in turn, allow them to provide numerous services such as infrastructure, software, and platforms.

Virtualization is a capability that allows different organisations or users to share the physical instance of a single application or resource. This strategy involves giving all of those physical resources a logical name and providing a reference to those physical resources based on demand.

We usually establish a virtual machine on top of an existing operating system and hardware, and then run additional operating systems or applications on top of it. Hardware virtualization is the term for this. The virtual machine creates a different environment that is logically separate from the hardware it runs on. The host machine is the system or machine, and the virtual machine is the guest machine. The firmware, referred to as a hypervisor, is in charge of managing this virtual environment.

Virtualization is an important part of cloud technology and its operation. In most cases, what happens in the cloud is that users not only share data stored in cloud-like applications, but they also share their infrastructures via virtualization. Virtualization is mostly utilised to provide cloud clients with standard versions of apps. The providers can efficiently supply the latest version of an application to the cloud and its users with the release of the latest version of that programme, and this is feasible using simply virtualization.

The system uses the cloud to manage network devices deployed on-premises at different locations. The solution requires Cisco Meraki cloud-managed devices, which provide full visibility of the network.

Cloud Networking is when all of an organization’s networking resources are hosted in the cloud. It can be either public or private, where a company can host. Cloud networking services are unique in relation to customary undertaking network plans. It is an application-based software infrastructure that stores data on serves that can be accessed through the internet using various front and back end data storage.

TYPES:

We need to identify the type of cloud-managed networking on which our cloud will be implemented. These are different types of cloud networking.

Public Cloud - which provides both services and infrastructure which is shared by all customers.

Private Cloud – which is utilized by a single organization.

Hybrid Cloud - This is a combination of both public and private cloud networks. It allows two platforms to interact for smooth functioning with data stored safely behind the firewalls.

MERAKI CLOUD - The Meraki cloud solution is a centralized management service that allows users to manage all of their Meraki network devices via a single simple and secure platform. Once a user makes a configuration change, the change request is sent to the Meraki cloud and is then pushed to the relevant device.

COMMUNICATION:

If a device is offline, it will continue to attempt to connect to the Meraki cloud until it gains connectivity. Once the device comes online, it automatically receives the most recent configuration settings from the Meraki cloud. If changes are made to the device configuration while the device is online, the device RECEIVES and updates these changes automatically. These changes are generally available on the device quickly. However, large quantities of changes may take longer time to reach. If no configuration changes are made by the user, the device continues to periodically check for updates to its configuration on its own. As the device runs on the network, it will communicate device and network usage analytics back to the Meraki cloud.

FEATURES:

·       Consistent and replicable configuration

·       Automatic firmware upgrades

·       Secure Site-to-Site VPN without previous IPsec knowledge - AutoVPN will automatically build secure IPsec tunnels between them.

·       Layer 7 traffic visibility – Meraki devices can filter or report traffic on your network based on application level.

·       Virtual Stacking – All Meraki switches support Virtual Stacking which lets us manage all switch ports as if there were all on a single switch. This rapidly reduces configuration effort.

It is used to Create cloud native applications store, back up and recover data

Stream audio and video etc.

**How is cloud related to your application?**

Normally in our application cloud will be helpful in storing the information like executed data, verified and not verified data and received data. Here the data can be a description about complaints and their proof for a problem. It can be accessed by any authoritative to know the status of the complaints and their execution.