**19CSE301 - Computer Networks CASE STUDY**

**Group - 5**

**AIRPORT Management System**

**GROUP DETAILS:**

|  |  |  |
| --- | --- | --- |
| **Group Member Roll No** | **Name of the Student** | **Department** |
| CB.EN.U4CSE19104 | A.V Siddhardha | Luggage Department |
| CB.EN.U4CSE19106 | Ayush Barik | Control Department |
| CB.EN.U4CSE19110 | D Monish | Maintenance Department |
|  |  |  |
|  |  |  |

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

|  |  |  |
| --- | --- | --- |
| **Department Name** | **Purpose** | **Network Details**  **(No of nodes, servers, Protocols)** |
| **Maintenance** | **To maintain serviced planes** | **Nodes-8, No of Servers –5, Protocols- (SMTP, FTP, HTTP)** |
| **Booking** | **Booking and Cancellation of flight tickets** | **Nodes-8, No of Servers –5, Protocols- (SMTP, FTP, HTTP)** |
| **Flights Control** | **Management of operations of services, flights, passengers and timings in the airport** |  |
|  |  |  |
|  |  |  |

**Case study:**

|  |  |  |
| --- | --- | --- |
| **Topic** | **Page No in the document** | **Supporting file name** |
| Problem Statement | **2** |  |
| Objective of the case study | **2** |  |
| Department Details in the case study with description | **4** |  |
| Network Architecture Diagram | **5** |  |
| Performance Parameters | **6** |  |
| Subnet ip scheme followed  [Solving the IP address allocation process for each subnetwork ] | **9** | **Ip\_scheme.jpeg** |
| Socket programming | **10** | **maintain-c.py,**  **maintain-s.py,**  **Airplace-Maintenance.csv** |
| Cisco packet tracer – Application layer protocols | **17** |  |
| Cisco packet tracer – Virtual Local Area Network | **18** | **19110\_vlan.pkt** |
| Cisco packet tracer – OSPF | **19** | **OSPF. pkt** |
| Cisco packet tracer – RIP | **19** | **RIP. pkt** |
| GoBack N and Selective Repeat protocol | **20** | **Server.java**  **client.java** |
| Cloud concepts | **26** |  |
| Cloud and Cisco packet tracer | **26** |  |
| Cloud and Networking | **26** |  |
| How is cloud related to your application? | **27** |  |

**Description of the case study:**

In this Application We are studying structure of Airport Network. The airport’s network provides services for different people, including passengers and staff Computer networking is the most crucial part of modern airports because this new technology takes the most important responsibilities, rather than people doing the tasks as in previous decades.

**The following thesis sheds light on some main parts which are improved during the practical work**:

* Maintaining passengers’ information
* Supporting the flight management system
* Maintenance Of Planes

**Why Networking is required For Our Application:**

Efficient communication within different sectors of an airport is essential, with highest priority being given to speed, efficiency and integrity of the information being passed. To enable that, a highly efficient networking solution becomes integral to the working of such a large infrastructural system where there could be a large cost involved due to the smallest of errors, and may even be fatal to hundreds of lives.

And most importantly By Network Airports deliver better communications Between every department, faster operations, increases safety and security

**Type of network:**

In Airport management system, we will be using MAN (Metropolitan area network). In this usecase, it is a well suited network type considering the scale of network required for this. Normally, airport networks use a LAN system for network connection between different buildings, but if we consider an airport network system of a metropolitan city domestic airport, the coverage is more than LAN systems, but less than a WAN variant. Hence, going the middle route of WAN, we satisfy both the conditions of wide range coverage for a network spanning multiple buildings, terminals and service points throughout the airport.

**Servers Used In Airport Management :**

a) Email server

b) FTP server

c) DNS server

**Network cable: coaxial cable** because fiber optic cable used for longer distances to transmit data in light speed but it is costly

Whereas coaxial cable Coaxial cable is used as a [transmission line](https://en.wikipedia.org/wiki/Transmission_line) for radio frequency signals we transmit signals in most of airports and in MAN’s we generally use coaxial cable and it has a data rate of 600Mbps

**Type of Topology**: **Hybrid**

**Different Departments Within The Application :**

**1) Bookings and Passenger Management :**

In This Department The Ticket Bookings And Ticket Cancellation of Passengers Takes place . These Are Managed By A Manager.In This Department we Will have 3 Systems Under Bookings Section and 1 Systen Under Management . So Regarding Flight Bookings Everything Will Be Done Here.

**2) Maintainance :**

In This We Will Maintain Every Thing Regarding Airplanes In The Airport Like Equipment (Engines,Sensors …. Every Part Of Plane Is Working Well Or Not Before Use ) And Central Manager Which Monitors Everything. Here For Equipment We Are Having 4 Systems and 1 For Central Manager

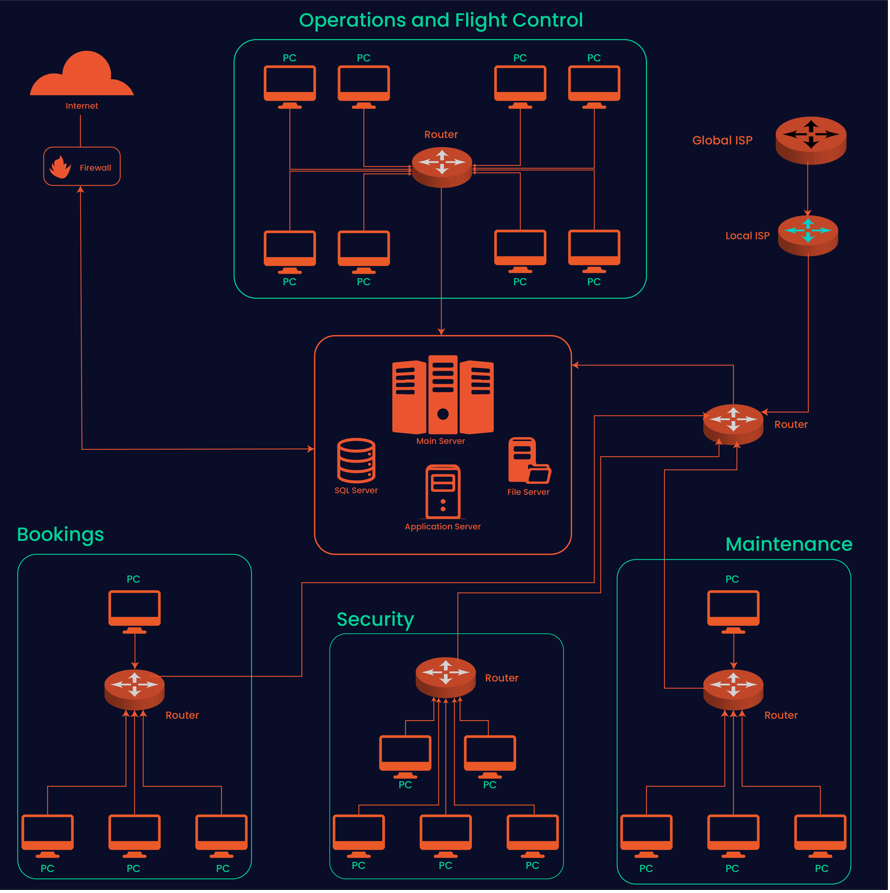
**3) Operations And Flight Control :**

he operations and flight control is responsible for four sectors - Airspace, Runway and Terminal, Services and Scheduling. Being the main control space, it requires high efficiency and accurate network coverage with no space for errors, as even small errors can cause huge losses.

Airspace control is required for communication with the in-air flights requiring landing and right after take offs to monitor their conditions, runway and terminals work alongside scheduling for docking, and assigning flights 6 their time and terminal docks.

Services are responsible for two stations - flight services such as fueling, damage checks, luggage loading/unloading, and crew-passenger services which are responsible for managing crew onland and in-air crew members, passenger and baggage loading, in-flight services such as food and special requests such as high priority passengers and specially abled.

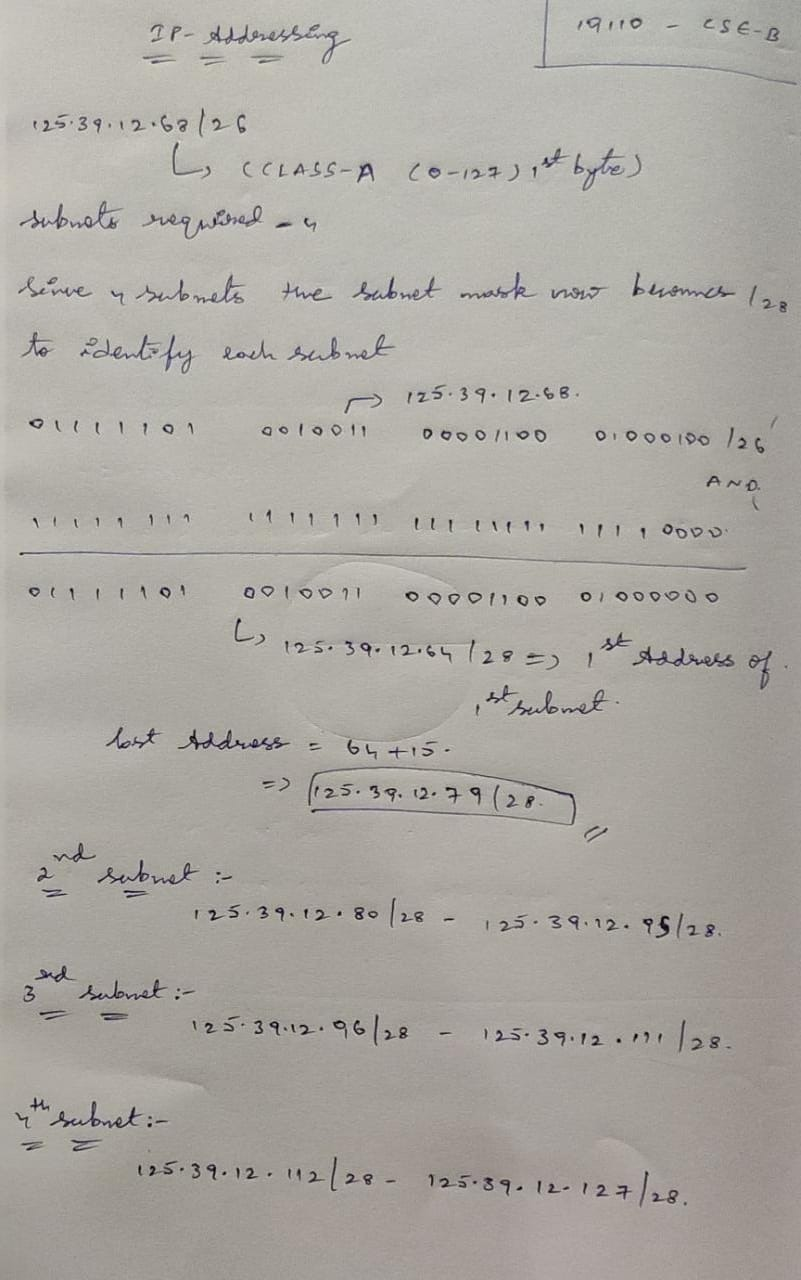
**Architecture Diagram:**



**Network Performance parameters with their purpose and formula:**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Meaning** | **Formula** |
| **Bandwidth** | Bandwidth is the capacity of a wired or wireless network communications link to transmit the maximum amount of data from one point to another over a computer network or internet connection in a given amount of time | Expressed as [bits](https://web.archive.org/web/20190816003233/https:/whatis.techtarget.com/definition/bit-binary-digit) per second ([bps](https://web.archive.org/web/20190816003233/https:/searchnetworking.techtarget.com/definition/bits-per-second)), modern network links have greater capacity, which is typically measured in millions of bits per second ([megabits per second](https://web.archive.org/web/20190816003233/https:/searchnetworking.techtarget.com/definition/Mbps), or Mbps) or billions of bits per second ([gigabits per second](https://web.archive.org/web/20190816003233/https:/whatis.techtarget.com/definition/Gbps-billions-of-bits-per-second), or Gbps). |
| **Throughput** | Throughput measures 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. |  |
| **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 |
| **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. | Directly proportional to processing speed of the routers. |
| **Queuing Delay** | Time spent by the data packet waiting in the queue before it is taken for execution is called queuing delay. | Directly Proportional to the congestion in the network |

**IP Addressing Scheme:**



**Department name : Airplane Maintenance**

**(cb.en.u4cse19110) Monish**

**Maintenance server can run more than one task simultaneously**

**Server contains all Operations needed and client request the server to perform those**

In this we maintain Plane parts

LM: (light maintenance) in this small parts are maintained and they are checked regularly

HM: (Heavy maintenance) checked every particular time large parts and important parts like Engines,cockpit etc …

Server have fllowing operations:

* In server we can add new plane and its corresponding values and **(I)**
* we can view details by requesting server from client **(V)**
* And to know how many planes are Not Checked(not ready for use)**(N)**
* And to know which planes are not checked **(U)**

**Server code: (maintain-s.py)**

import socket

import csv

import pandas as pd

import numpy as np

from csv import writer

from \_thread import \*

df = pd.read\_csv("C:/Users/monis/Desktop/Airplane-maintenance.csv")

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

try:

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

except socket.error as error:

print("Sorry ! Check Your Connection" + str(error))

print ("Maintenance Server running")

print("Waiting For Connection ........ ")

s.listen(10)

list1=[]

def threaded\_client(clt):

while True:

ch = clt.recv(10).decode("utf-8")

if(ch == 'I'):

n = clt.recv(10).decode("utf-8")

for i in range(1, int(n)+1):

clt.send(bytes("Enter Flight-name :", "utf-8"))

fn = clt.recv(1000).decode("utf-8")

clt.send(bytes("MAX load ", "utf-8"))

load = clt.recv(1000).decode("utf-8")

clt.send(bytes("No of Sensors: ", "utf-8"))

Sensors = clt.recv(1000).decode("utf-8")

clt.send(bytes("No of Engines: ", "utf-8"))

engines = clt.recv(1000).decode("utf-8")

clt.send(bytes("LM ?:", "utf-8"))

lm = clt.recv(1000).decode("utf-8")

clt.send(bytes("HM ?:", "utf-8"))

hm = clt.recv(1000).decode("utf-8")

clt.send(bytes("No of seats:", "utf-8"))

seats = clt.recv(1000).decode("utf-8")

clt.send(bytes("Checked ? :", "utf-8"))

check = clt.recv(1000).decode("utf-8")

list1=[fn,load,Sensors,engines,lm,hm,seats,check]

with open("C:/Users/monis/Desktop/Airplane-maintenance.csv",'a', newline='') as csvfile:

writer = csv.writer(csvfile)

writer.writerow(list1)

elif(ch == 'V'):

clt.send(bytes("C:/users/monis/Desktop/Airplane-maintenance","utf-8"))

elif(ch == 'N'):

a=df[df['check'] == 'No']['F N'].count()

a=a.item()

clt.send(repr(a).encode("utf-8"))

elif(ch == 'U'):

rslt = df.loc[df['check'] == 'No']

a=rslt['F N']

clt.send(str(a).encode("utf-8"))

while True:

clt, adr = s.accept()

print(f"Connected Successfully to {adr} Client")

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

clt.send(bytes("Give Following Inputs\nI for Insertion\nV for View\nN To count Nnumber Planes are checked\nU To Check Planes That are not serviced\n", "utf-8"))

**client code: (maintain-c.py)**

import socket

import pandas as pd

import numpy as np

import csv

df = pd.read\_csv(r"C:/Users/monis/Desktop/Airplane-maintenance.csv")

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

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

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

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

if(ch == 'I'):

n = int(input("Enter number of rows to be inserted :"))

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

for i in range(1, n + 1):

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

elif(ch == 'V'):

view=(s.recv(5000).decode("utf-8"))

print(view)

elif(ch=='N'):

print("Number Of Planes Unchecked Are = ",s.recv(1000).decode("utf-8"))

elif(ch=='U'):

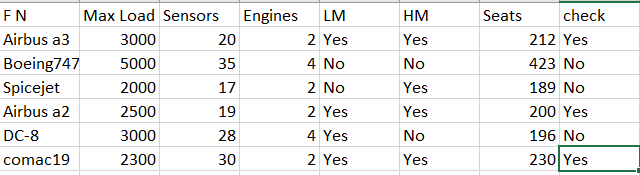
print("Planes Unchecked Are : ")

unchecked\_planes=s.recv(1000).decode("utf-8")

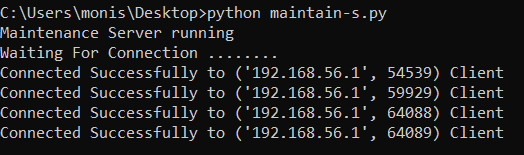
print(unchecked\_planes)

**outputs:**

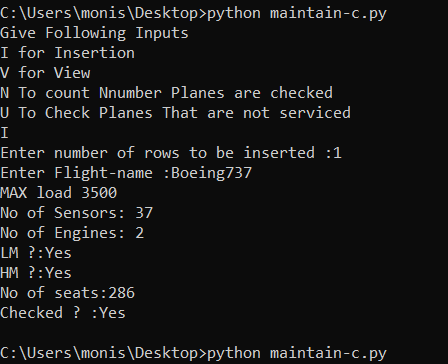
Before adding New plane:



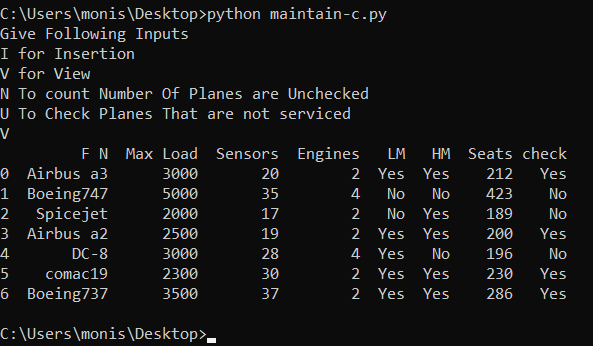
**Output in server side:**



**Outputs on client side:**

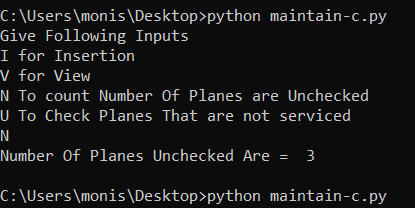


After again if we run client and give input as V

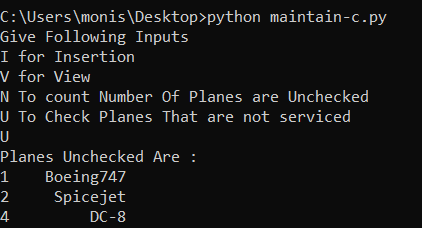


We can clearly see that it is updated

**To see How many planes are not serviced**



**Planes names that are Not checked (serviced):**



**Application layer protocols :**

a.) File Transfer Protocol (FTP)

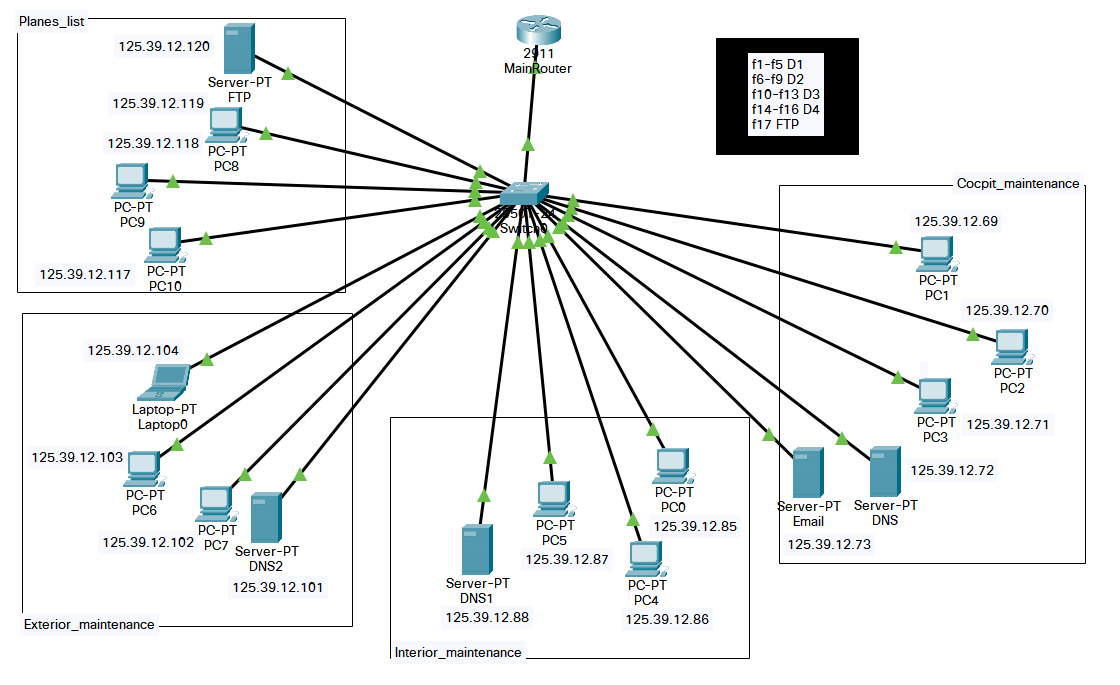
b.) Simple Mail Transfer Protocol (SMTP)

c.) Hyper Text Transfer Protocol (HTTP)

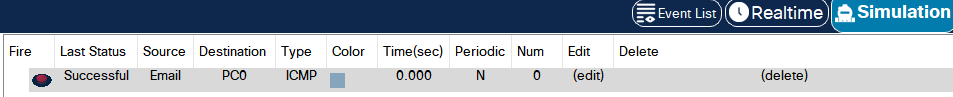
**Implementing Routing Protocols In Packet Tracer In Maintenance Department:**

**1.**

**Virtual LAN:**

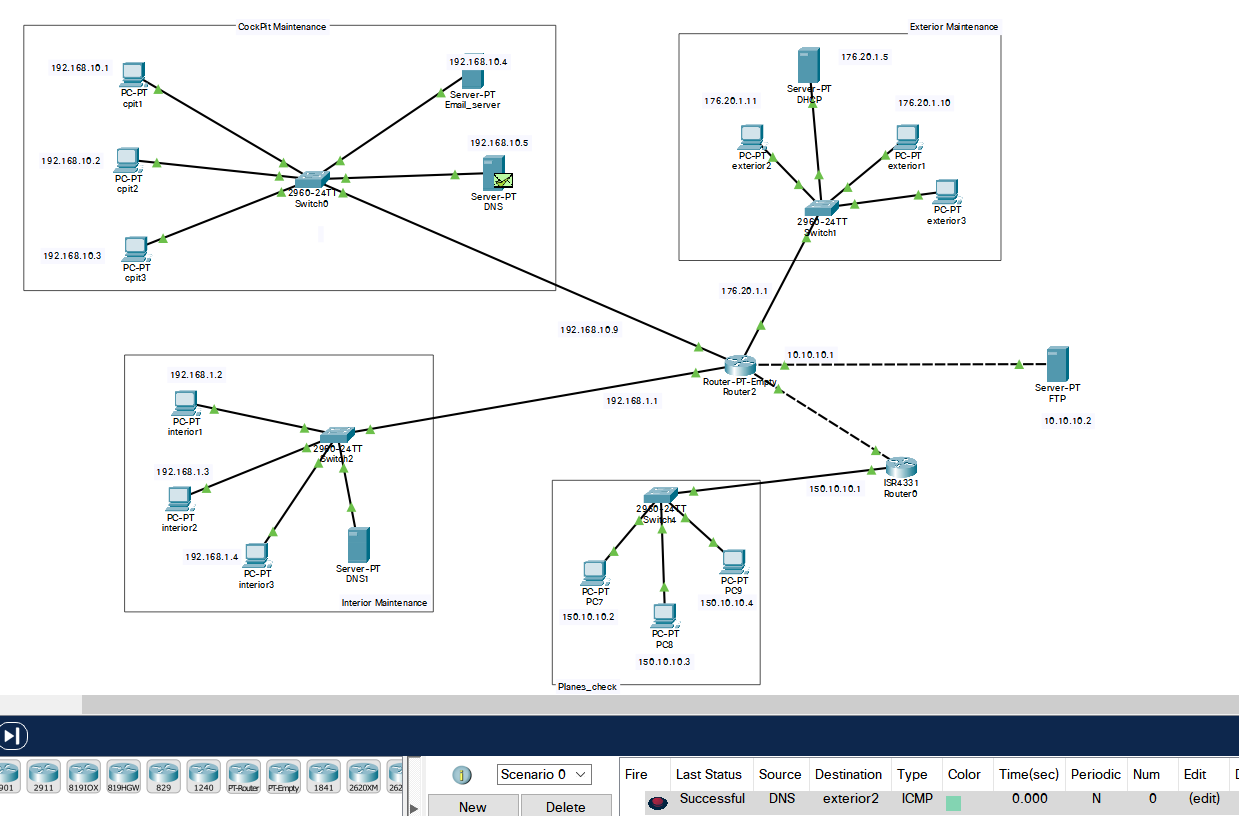


**Working Of VLAN:**

****

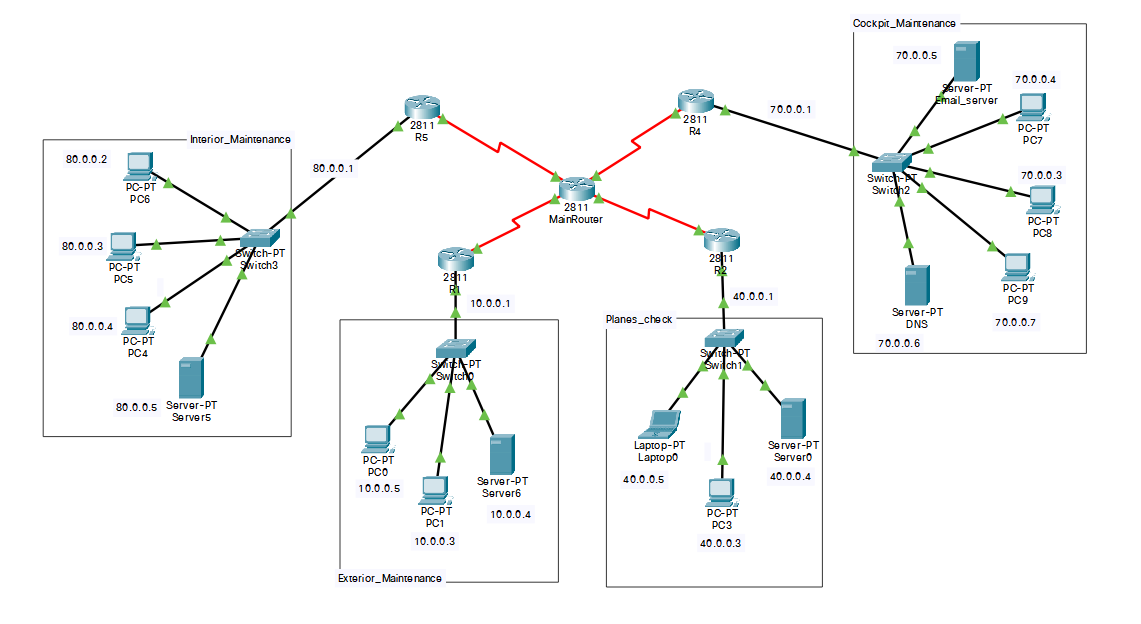
**2.**

**RIP And Its Working:**

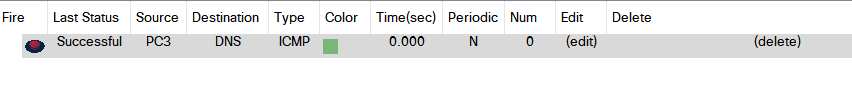


**3.**

**OSPF:**

****

**Working Of OSPF:**

****

**GoBackN and Selective Repeat:**

**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[10];

//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 FRAMES 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 THE 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);

}

}

}

**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, 75, 80, 90, 100, 110 };

Serversocket = new ServerSocket(8011);

System.out.println("WAITING FOR A 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();

}

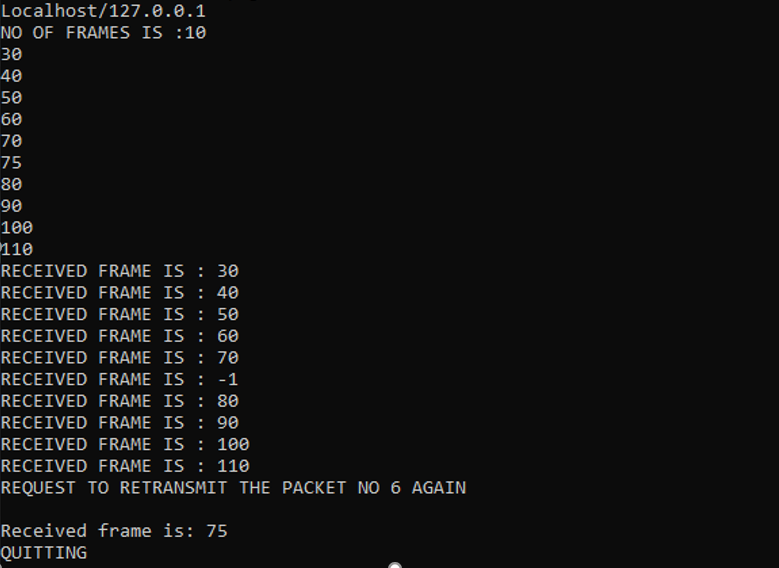
}

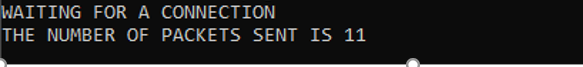
}

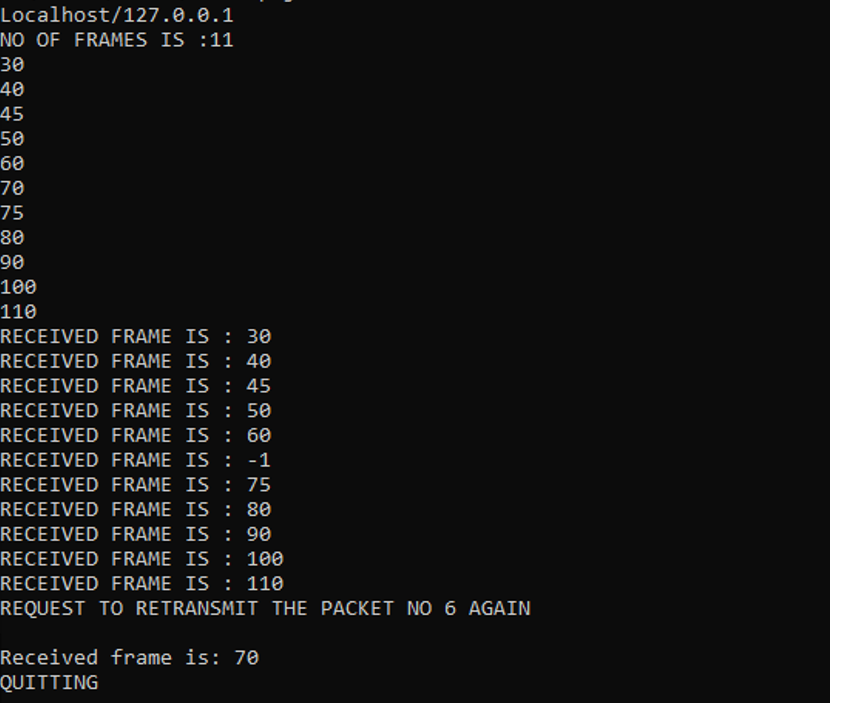
}

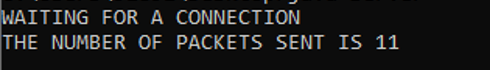
**OUTPUT**

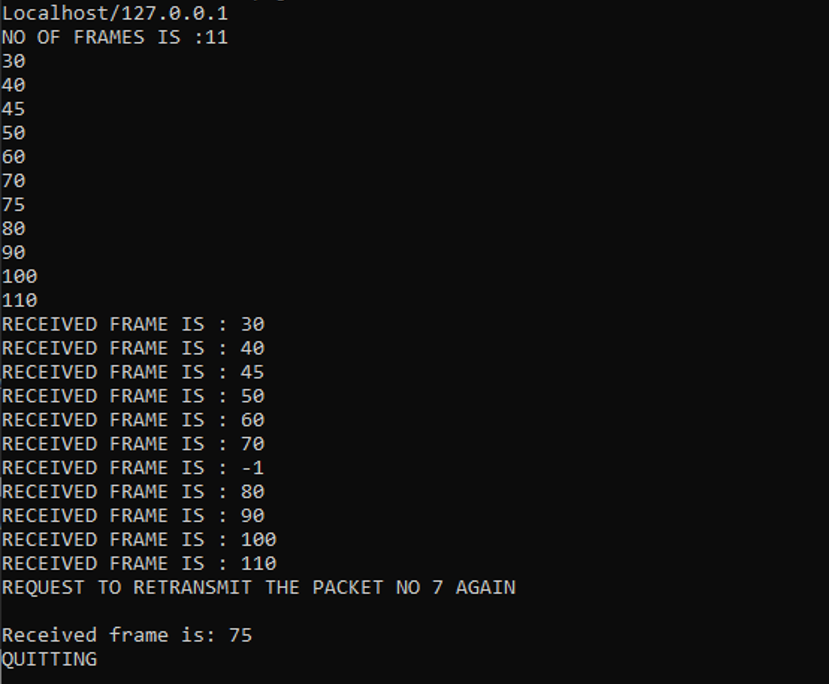












**Cloud Visualization**

Virtualization is a technique of how to separate a service from the underlying physical delivery of that service.

It was initially developed during the mainframe era. It involves using specialized software to create a virtual or software-created version of a computing resource rather than the actual version of the same resource. With the help of Virtualization, multiple operating systems and applications can run on same machine and its same hardware at the same time, increasing the utilization and flexibility of hardware.

In other words, one of the main cost effective, hardware reducing, and energy saving techniques used by cloud providers is virtualization. The term virtualization is often synonymous with hardware virtualization, which plays a fundamental role in efficiently delivering Infrastructure-as-a-Service (IaaS) solutions for cloud computing.

Moreover, virtualization technologies provide a virtual environment for not only executing applications but also for storage, memory, and networking.

## **Virtualization Concept**

Creating a virtual machine over existing operating system and hardware is referred as Hardware Virtualization. Virtual Machines provide an environment that is logically separated from the underlying hardware.

The machine on which the virtual machine is created is known as host machine and virtual machine is referred as a guest machine. This virtual machine is managed by a software or firmware, which is known as hypervisor**.**

**Difference Between Virtualization and Cloud**

1. Essentially there is a gap between these two terms, though cloud technology requires the concept of virtualization. Virtualization is a technology - it can also be treated as software that can manipulate hardware
2. Virtualization is the foundation element of cloud computing, whereas Cloud technology is the delivery of shared resources as a service-on-demand Through the internet.
3. Cloud is essentially made-up of the concept of virtualization.

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

In our project even though information about flights and passengers can be stored on in FTP - servers, storing on a cloud based database is much more preferable because

1. Cloud is Highly scalable
2. More secure due to third party managing the actual data
3. Information easily accessible anywhere