

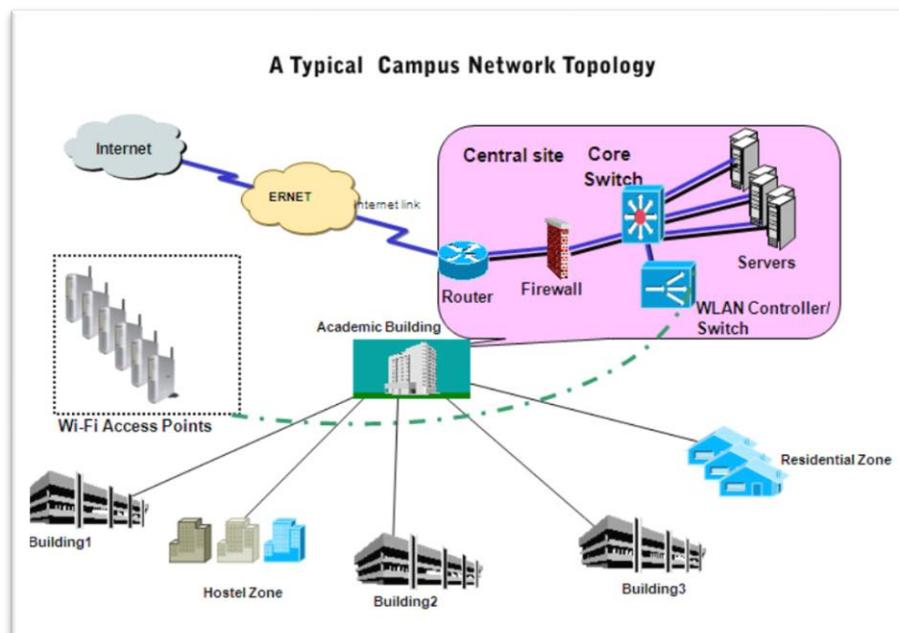
Title : University Network Management

Roll No.	Name	Contribution in case study
CB.EN.U4CSE19346	Raghul K B	Cloud Computing Virtualization
CB.EN.U4CSE19358	V Devakumar	Socket Programming
CB.EN.U4CSE19359	V Neelesh Gupta	Documentation
CB.EN.U4CSE19360	V Nithin Krishna	Implementation in Cisco

DESCRIPTION :

Campus networking is rightly perceived by most colleges and universities to be one of the most important issues facing them at present. A campus network is far more than just a physical infrastructure, and planning and building a campus network really means developing an entirely new information environment that will have a profound impact on almost every aspect of campus life. In building networks, campuses are engaged in a change process that is strategically essential to their very nature and existence. The network that we've built offers some advice to administrators and faculty about things to consider as they engage in planning, building, and managing campus networks in small college environments.

It is important to understand that network planning really means planning a whole new environment that will encompass information, communication, and entertainment on the campus. The network will provide a powerful, dynamic, new channel of access to the world of information. It will also enhance and fundamentally alter the communication mechanisms on campus. And whether we want it or not, the network will provide a wide range of new entertainment opportunities. This new environment will encompass or touch all the information technology areas (e.g., computing, library automation, audio, video and multimedia technologies), which will challenge, compete with, and sometimes replace existing modes of campus communications, information access, administrative work, power hierarchies, and budgeting systems.



WHY IS NETWORKING REQUIRED :

In today's world, personal computers and computer networking are an essential aspect of communication and productivity. The Internet is a crucial aspect of computer networking that is widely used in a variety of industries, including business and higher education. Students and professors at colleges and universities can benefit from the Internet in a variety of ways.

The Internet helps to speed the college and university admissions process. Many schools offer online application downloads, and some even have online application forms. Prospective students can usually find information on university websites to assist them make an informed decision about which school to attend.

The Internet is a valuable resource for students and educators performing research. When compared to searching for the same information on a computer, going to a library and searching through a card catalogue by hand can be time-consuming and inefficient. Many colleges have online library systems that allow students to use lab computers to look up information in books or access databases of scholarly publications to read online.

The ability to communicate swiftly with huge groups of individuals is an advantage of using the Internet in university education. Students and professors can communicate via email even if they are unable to meet in person. It allows educators to send out announcements like assignments or course syllabi without having to distribute printed copies. Students can submit assignments by email or another online submission system, reducing the quantity of paper trash generated by the university. Students who must skip class can also benefit from online communication because they can obtain class notes and other vital information.

The Internet's role in e-Learning is one of the most powerful advantages of the Internet in higher education. E-learning is a method of teaching that relies solely on computer or Internet-based materials. Some colleges provide entirely e-learning courses, which means that all textbooks, assignments, quizzes, and examinations are available online with no need for face-to-face sessions. Students with limited mobility or tight schedules can use e-learning to take courses and study at their leisure. E-learning also allows students to learn at their own pace; one student may complete a course in half the time that another student does. Many colleges and universities offer online courses or have courses with aspects of e-learning, such as online quizzes and exams.

References : <https://ernet.in/content/campus-wide-networking>

<https://www.theclassroom.com/effects-computers-college-3982.html>

Textbook – Computer Networking A Top-Down Approach 7th Edition

NETWORK PERFORMANCE PARAMETERS :

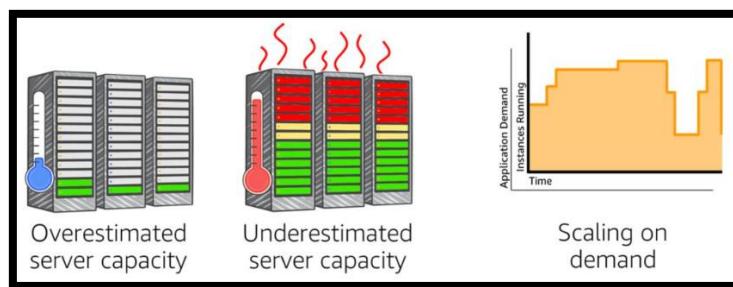
Parameter	Definition	Formula and Values
Throughput	Throughput is a measure of how many units of information a system can process in a given amount of time. It measures your network's actual data transmission rate, which can vary wildly through different areas of your network.	Network throughput is measured in bits per second (bps). Maximum network throughput equals the TCP window size divided by the round-trip time of communications data packets. $\text{Throughput} \leq \frac{\text{RWIN}}{\text{RTT}}$ <p>RWIN = TCP receive window (max 65535 bytes), RTT = Round trip time(in seconds)</p>
Packet Loss	Packet loss refers to the number of data packets that were successfully sent out from one point in a network, but were dropped during data transmission and never reached their destination. The more data packets that are lost, the longer it takes for a data request to be fulfilled	$\text{PLR} = \frac{N^{tx} - N^{rx}}{N^{tx}} \times 100\%$ <p>PLR = Packet Loss Rate Ntx = total number of transmitted packets Ntr = total number of received packets</p>
Bandwidth	The maximum amount of data transmitted over an internet connection in a given amount of time. It's the volume of information that can be sent over a connection in a measured amount of time.	Bandwidth are expressed in metric prefixes, such as Mbps, Gbps etc. Bandwidth can also be expressed as bytes per second.
Latency	Latency refers to the measure of time it takes for data to reach its destination across a network. A computer that uses a TCP/IP network sends a limited amount of data to its destination and then waits for an acknowledgement that the data has reached its destination before sending any more. This round trip delay (latency) has a big impact on network performance.	Measured in milliseconds.

Propagation Delay	Propagation delay is the amount of time it takes for the head of the signal to travel from the sender to the receiver. Propagation delay refers to the amount of time it takes for the first bit to travel over a link between sender and receiver, whereas latency refers to the total amount of time it takes to send an entire message.	Network propagation delay is measured in nanoseconds (nS). It can be computed as the ratio between the link length and the propagation speed over the specific medium. Propagation Delay = Link length/Propagation speed
Transmission Time	Transmission time is the amount of time from the beginning until the end of a message transmission. In the case of a digital message, it is the time from the first bit until the last bit of a message has left the transmitting node.	The packet transmission time in seconds can be obtained from the packet size in bit and the bit rate in bit/s as: Packet transmission time = Packet size / Bit rate
Jitter	Information is transported from your computer in data packets across the internet. They are usually sent at regular intervals and take a set amount of time. Jitter is when there is a time delay in the sending of these data packets over your network connection. This is often caused by network congestion, and sometimes route changes. Simply put, jitter is a variation in delay.	Add all the difference between Latencies, then divide by the number of samples (minus 1).
Packet Reordering	It occurs when data packets arrive at their destination in the wrong order. This can happen for various reasons, such as multi-path routing, route fluttering, and wrong QoS queue configuration.	
Packet Duplication	Packet duplication refers to when data packets are duplicated somewhere in the network, and are then received twice at their destination.	

CLOUD COMPUTING VIRTUALIZATION :

In Cloud Computing, virtualization allows a provider to virtualize servers, storage, or other physical hardware or data centre resources, allowing them to provide a variety of services including infrastructure, software, and platforms. Users can utilise infrastructure as a service (IaaS) to access cloud-based servers, storage, and network resources: a whole infrastructure available via cloud virtualization technologies. It means that consumers don't have to build or buy their own infrastructure; instead, they can use the virtualized version. Most people have heard of Software as a Service (SaaS), which is software that is abstracted from hardware and accessed via the cloud.

In this University Network you have a server for all the services that you run in the university i.e., Conducting Quizzes, Access the FTP, etc..., With the Cloud You Eliminate guessing about your infrastructure capacity needs. When you decide on capacity before deploying an application, you typically end up with either expensive idle resources or constrained capacity. These issues are no longer an issue because to cloud computing. You have access to as much or as little as you need, and you may scale up or down with just a few minutes' notice.



Benefits of Cloud over Traditional Servers

- Trade capital expense for variable expense

Capital expenses (capex) are funds that a company uses to acquire, upgrade, and maintain physical assets such as property, industrial buildings, or equipment. Instead of investing heavily in data centers and servers before you know how you will use them, you can pay only when you consume resources and pay only for the amount you consume. Thus, you save money on technology. It also enables you to adapt to new applications with as much space as you need in minutes, instead of weeks or days. Maintenance is reduced, so you can spend focus more on the core goals of your business.

- Massive economies of scale

By using cloud computing, you can achieve a lower variable cost than you can get on your own. Because usage from hundreds of thousands of customers is aggregated in the cloud, providers such as AWS can achieve higher economies of scale, which translates into lower pay-as-you-go prices.

- Stop guessing capacity

Eliminate guessing about your infrastructure capacity needs. When you make a capacity decision before you deploy an application, you often either have expensive idle resources or deal with limited capacity. With cloud computing, these problems go away. You can access as much or as little as you need, and scale up and down as required with only a few minutes' notice.

- Increase speed and agility

In a cloud computing environment, new IT resources are only a click away, which means that you reduce the time it takes to make those resources available to your developers from weeks to just minutes. The result is a dramatic increase in agility for the organization because the cost and time that it takes to experiment and develop are significantly lower.

- Stop spending money on running and maintaining data centers

Focus on projects that differentiate your business instead of focusing on the infrastructure. Cloud computing enables you to focus on your own customers instead of the heavy lifting of racking, stacking, and powering servers.

- Go global in minutes

You can deploy your application in multiple AWS Regions around the world with just a few clicks. As a result, you can provide a lower latency and better experience for your customers simply and at minimal cost.

List of some top cloud computing companies in India and across the world

- Amazon Web Service (AWS)
- ServerSpace.
- Microsoft Azure.
- Google Cloud Platform.
- IBM Cloud Services.
- Adobe Creative Cloud.
- Kamatera.
- VMware.

SOCKET PROGRAMMING :

Description of the records.xlsx file

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I	J	K	L
1	Roll Number	Date of Birth	Student Name	Gender	Branch	Joining Year	No of Arrears	Attendance(%)	CGPA			
2	CB.EN.U4CSE19346	22-Feb-2001	Raghul K B	Male	CSE	2019	0	96	8.35			
3	CB.EN.U4CSE19358	27-Aug-2002	Devakumar	Male	CSE	2019	0	91	8.45			
4	CB.EN.U4CSE19359	23-Apr-2001	Neelesh Gupta	Male	CSE	2019	0	78.23	8.12			
5	CB.EN.U4CSE19360	15-Jul-2001	Nithin Krishna V	Male	CSE	2019	0	95	8.65			
6												

Column Name	Description
Roll Number	Unique ID to identify a student
Date of birth	The Day, Month, Year that a student was born
Student Name	The Name of the Student
Gender	Gender of the student
Branch	Branch of Engineering that the student is in
Joining Year	Year that the student was admitted in the university
No of Arrears	Total No of Arrears that the student has
Attendance(%)	Overall Percentage of Attendance
CGPA	Cumulative Grade Point Average

List of Operations [SERVER – STUDENT]

UP – Update Profile (where student is able to update his details except Attendance(%), CGPA, Roll Number, and No of Arrears)

VP – View Profile (where student can view his/her profile)

List of Operations [SERVER – ADMIN]

I – Insert new record (where Admin has to enter complete information of student to enter it in the server)

U – Update a record (where Admin Updates a record by student roll number)

D – Delete a record (where Admin can delete a record by student roll number)

C – Copy all records (Copies all records into a new file named records_copy.xlsx)

V – View all records (where Admin can view all records present in the server)

CODE : [SERVER – STUDENT] – SINGLE SERVER MULTI CLIENT

server.py

```
import socket
from _thread import *
import pandas as pd

ServerSocket = socket.socket()
host = socket.gethostname()
port = 2004
ThreadCount = 0
try:
    ServerSocket.bind((host, port))
except socket.error as e:
    print(str(e))

print(f'Socket is listening on {host}')
ServerSocket.listen(5)

def multi_threaded_client(connection):
    connection.send(str.encode('Server is working:'))
    while True:
        msg = connection.recv(2048).decode('utf-8')
        data = msg.split('%')
        df = pd.read_excel("records.xlsx")
        records = df.to_dict(orient='list')

        if (data[0] == 'UP'):
            temp = records['Roll Number']
            for j in range(len(temp)):
                if temp[j] == data[1]:
                    records['Student Name'][j] = data[2]
                    records['Date of Birth'][j] = str(data[3])
                    records['Gender'][j] = data[4]
                    records['Joining Year'][j] = data[5]
                    records['Branch'][j] = data[6]
            new = pd.DataFrame.from_dict(records,
orient='index').transpose()
            new.to_excel('records.xlsx', index=False)
            send_msg = "Updated Successfully"

        elif (data[0] == 'VP'):
            send_msg = ""
            temp = records['Roll Number']
            lis = []
            for i in range(len(temp)):
                if temp[i] == data[1]:
                    lis.append(i)

            for j in records.keys():
                send_msg += j
                send_msg += "*"
                for k in lis:
                    send_msg += str(records[j][k])
                    send_msg += "%"
                send_msg += "*"

            if not data:
                break
```

```

        connection.sendall(str.encode(send_msg))
        connection.close()

while True:
    Client, address = ServerSocket.accept()
    print('Connected to: ' + address[0] + ':' + str(address[1]))
    start_new_thread(multi_threaded_client, (Client,))
    ThreadCount += 1
    print('Total Client Connections: ' + str(ThreadCount))
ServerSocket.close()

```

student.py

```

import socket
import pandas as pd

MultiClientSocket = socket.socket()
host = socket.gethostname()
port = 2004

print('Waiting for response')
try:
    MultiClientSocket.connect((host, port))
except socket.error as e:
    print(str(e))

res = MultiClientSocket.recv(1024)
while True:
    print(" UP - Update Profile")
    print(" VP - View Profile")
    Input = input("Choose from the above options: ")
    data = Input
    data += '%'
    if (Input == "UP"):
        print('Enter the Roll Number:', end="")
        temp = input()
        temp += '%'
        data += temp
        print('Enter the Name:', end="")
        temp = input()
        temp += '%'
        data += temp
        print('Enter Date of Birth:', end="")
        temp = input()
        temp += '%'
        data += temp
        print('Enter Gender:', end="")
        temp = input()
        temp += '%'
        data += temp
        print('Enter Joining Year:', end="")
        temp = input()
        temp += '%'
        data += temp
        print('Enter Branch:', end="")
        temp = input()
        data += temp

```

```
elif (Input == "VP"):  
    print('Enter the Roll Number:', end="")  
    temp = input()  
    data += temp  
  
MultiClientSocket.send(str.encode(data))  
res = MultiClientSocket.recv(1024)  
if Input == "UP":  
    print(res.decode('utf-8'))  
elif Input == "VP":  
    x = res.decode('utf-8').split('*')  
    fin = {}  
    for i in range(1, len(x)):  
        if (i % 2 != 0):  
            temp = x[i].split("%")  
            fin[x[i - 1]] = temp  
    output = pd.DataFrame.from_dict(fin)  
    size = len(output)  
    print(output.head(size - 1))  
  
MultiClientSocket.close()
```

OUTPUT : [SERVER – STUDENT] – SINGLE SERVER MULTI CLIENT

Intial records.xlsx file

	A	B	C	D	E	F	G	H	I
1	Roll Number	Date of Birth	Student Name	Gender	Branch	Joining Year	No of Arrears	Attendance(%)	CGPA
2	CB.EN.U4CSE19346	22-Feb-2001	Raghul K B	Male	CSE	2019	0	96	8.35
3	CB.EN.U4CSE19358	27-Aug-2002	Devakumar	Male	CSE	2019	0	91	8.45
4	CB.EN.U4CSE19359	13-May-2000	Neelesh V	Male	CSE	2019	0	93	8.55
5	CB.EN.U4CSE19360	15-Jul-2001	Nithin Krishna V	Male	CSE	2019	0	95	8.65

```
E:\>cd Python SP  
E:\Python SP>python server.py  
Socket is listening on 192.168.1.36  
Connected to: 192.168.1.36:63277  
Total Client Connections: 1
```

View Profile [only his/her profile]

```
E:\Python SP>python student.py  
Waiting for response  
UP - Update Profile  
VP - View Profile  
Choose from the above options: VP  
Enter the Roll Number:CB.EN.U4CSE19358  
Roll Number Date of Birth Student Name Gender Branch Joining Year No of Arrears Attendance(%) CGPA  
0 CB.EN.U4CSE19358 27-Aug-2002 Devakumar Male CSE 2019 0 91 8.45  
UP - Update Profile  
VP - View Profile  
Choose from the above options:
```

```
E:\Python SP>python server.py  
Socket is listening on 192.168.1.36  
Connected to: 192.168.1.36:63314  
Total Client Connections: 1  
Connected to: 192.168.1.36:63332  
Total Client Connections: 2
```

View Profile [only his/her profile]

```
E:\Python SP>python student.py  
Waiting for response  
UP - Update Profile  
VP - View Profile  
Choose from the above options: VP  
Enter the Roll Number:CB.EN.U4CSE19346  
Roll Number Date of Birth Student Name Gender Branch Joining Year No of Arrears Attendance(%) CGPA  
0 CB.EN.U4CSE19346 22-Feb-2001 Raghul K B Male CSE 2019 0 96 8.35  
UP - Update Profile  
VP - View Profile  
Choose from the above options:
```

```
E:\Python SP>python server.py
Socket is listening on 192.168.1.36
Connected to: 192.168.1.36:63390
Total Client Connections: 1
Connected to: 192.168.1.36:63391
Total Client Connections: 2
Connected to: 192.168.1.36:63400
Total Client Connections: 3
```

Update Profile [only his/her profile]

```
E:\Python SP>python student.py
Waiting for response
UP - Update Profile
VP - View Profile
Choose from the above options: UP
Enter the Roll Number:CB.EN.U4CSE19359
Enter the Name:Neelesh Gupta
Enter Date of Birth:13-Apr-2001
Enter Gender:Male
Enter Joining Year:2019
Enter Branch:CSE
Updated Successfully
UP - Update Profile
VP - View Profile
Choose from the above options:
```

Updated records.xlsx file

	A	B	C	D	E	F	G	H	I
1	Roll Number	Date of Birth	Student Name	Gender	Branch	Joining Year	No of Arrears	Attendance(%)	CGPA
2	CB.EN.U4CSE19346	22-Feb-2001	Raghul K B	Male	CSE	2019	0	96	8.35
3	CB.EN.U4CSE19358	27-Aug-2002	Devakumar	Male	CSE	2019	0	91	8.45
4	CB.EN.U4CSE19359	13-Apr-2001	Neelesh Gupta	Male	CSE	2019	0	93	8.55
5	CB.EN.U4CSE19360	15-Jul-2001	Nithin Krishna V	Male	CSE	2019	0	95	8.65
6									

```
E:\Python SP>python server.py
Socket is listening on 192.168.1.36
Connected to: 192.168.1.36:63390
Total Client Connections: 1
Connected to: 192.168.1.36:63391
Total Client Connections: 2
Connected to: 192.168.1.36:63400
Total Client Connections: 3
Connected to: 192.168.1.36:63435
Total Client Connections: 4
```

View Profile [only his/her profile]

```
E:\Python SP>python student.py
Waiting for response
UP - Update Profile
VP - View Profile
Choose from the above options: VP
Enter the Roll Number:CB.EN.U4CSE19360
      Roll Number Date of Birth      Student Name Gender Branch Joining Year No of Arrears Attendance(%) CGPA
0   CB.EN.U4CSE19360  15-Jul-2001  Nithin Krishna V  Male    CSE        2019          0         95     8.65
UP - Update Profile
VP - View Profile
Choose from the above options:
```

CODE : [SERVER - ADMINISTRATOR] – SINGLE SERVER MULTI CLIENT

server.py

```
import socket
from _thread import *
import pandas as pd

ServerSocket = socket.socket()
host = socket.gethostname()
port = 2004
ThreadCount = 0
try:
    ServerSocket.bind((host, port))
except socket.error as e:
    print(str(e))

print(f'Socket is listening on {host}')
ServerSocket.listen(5)

def multi_threaded_client(connection):
    connection.send(str.encode('Server is working:'))
    while True:
        msg = connection.recv(2048).decode('utf-8')
        data = msg.split('%')
        df = pd.read_excel("records.xlsx")
        records = df.to_dict(orient='list')
        if (data[0] == 'I'):
            records['Roll Number'].append(data[1])
            records['Date of Birth'].append(str(data[2]))
            records['Student Name'].append(data[3])
            records['Gender'].append(data[4])
            records['Branch'].append(data[5])
            records['Joining Year'].append(int(data[6]))
            records['No of Arrears'].append(int(data[7]))
            records['Attendance(%)'].append(float(data[8]))
            records['CGPA'].append(float(data[9]))
            new = pd.DataFrame.from_dict(records,
orient='index').transpose()
            new.to_excel('records.xlsx', index=False)
            send = "Inserted Successfully"

        elif (data[0] == 'U'):
            temp = records['Roll Number']
            for j in range(len(temp)):
                if temp[j] == data[1]:
                    records['Date of Birth'][j] = str(data[2])
                    records['Student Name'][j] = data[3]
                    records['Gender'][j] = data[4]
                    records['Branch'][j] = data[5]
                    records['Joining Year'][j] = int(data[6])
                    records['No of Arrears'][j] = int(data[7])
                    records['Attendance(%)'][j] = float(data[8])
                    records['CGPA'][j] = float(data[9])
            new = pd.DataFrame.from_dict(records,
orient='index').transpose()
            new.to_excel('records.xlsx', index=False)
            send = "Updated Successfully"

        elif (data[0] == 'D'):
```

```

temp = records['Roll Number']
for j in range(len(temp)):
    if temp[j] == data[1]:
        del records['Roll Number'][j]
        del records['Date of Birth'][j]
        del records['Student Name'][j]
        del records['Gender'][j]
        del records['Branch'][j]
        del records['Joining Year'][j]
        del records['No of Arrears'][j]
        del records['Attendance(%)'][j]
        del records['CGPA'][j]
    new = pd.DataFrame.from_dict(records,
orient='index').transpose()
    new.to_excel('records.xlsx', index=False)
    send = "Deleted Successfully"

elif (data[0] == 'C'):
    copyx = pd.DataFrame.from_dict(records,
orient='index').transpose()
    copyx.to_excel('records_copy.xlsx', index=False)
    send = "Copied Successfully"

elif (data[0] == "V"):
    send = ""
    for j in records.keys():
        send += j
        send += "*"
        for k in records[j]:
            send += str(k)
            send += "%"
        send += "*"

    if not data:
        break
    connection.sendall(str.encode(send))
connection.close()

while True:
    Client, address = ServerSocket.accept()
    print('Connected to: ' + address[0] + ':' + str(address[1]))
    start_new_thread(multi_threaded_client, (Client,))
    ThreadCount += 1
    print('Total Client Connections: ' + str(ThreadCount))
ServerSocket.close()

```

admin.py

```

import socket
import pandas as pd

MultiClientSocket = socket.socket()
host = socket.gethostname()
port = 2004

print('Waiting for response')
try:

```

```
        MultiClientSocket.connect((host, port))
    except socket.error as e:
        print(str(e))

    res = MultiClientSocket.recv(1024)
    while True:
        print(" I - Insert new record")
        print(" U - Update a record")
        print(" D - Delete a record")
        print(" C - Copy all records")
        print(" V - View all records")
        Input = input("Choose from the above options: ")
        data = Input
        data += '%'
        if (Input == 'I'):
            print("Enter Roll Number:", end="")
            temp = input()
            temp += '%'
            data += temp
            print("Enter Date of Birth:", end="")
            temp = input()
            temp += '%'
            data += temp
            print("Enter Student Name:", end="")
            temp = input()
            temp += '%'
            data += temp
            print("Enter Gender:", end="")
            temp = input()
            temp += '%'
            data += temp
            print("Enter Branch:", end="")
            temp = input()
            temp += '%'
            data += temp
            print("Enter Joining Year:", end="")
            temp = input()
            temp += '%'
            data += temp
            print("Enter No. of Arrears:", end="")
            temp = input()
            temp += '%'
            data += temp
            print("Enter Attendance(%):", end="")
            temp = input()
            temp += '%'
            data += temp
            print("Enter CGPA:", end="")
            temp = input()
            data += temp

        elif (Input == "D"):
            print('Enter Roll Number:', end="")
            temp = input()
            data += temp

        elif (Input == "U"):
            print('Enter Roll Number:', end="")
            temp = input()
            temp += '%'
            data += temp
```

```

print("Enter Date of Birth:", end="")
temp = input()
temp += '%'
data += temp
print("Enter Student Name:", end="")
temp = input()
temp += '%'
data += temp
print("Enter Gender:", end="")
temp = input()
temp += '%'
data += temp
print("Enter Branch:", end="")
temp = input()
temp += '%'
data += temp
print("Enter Joining Year:", end="")
temp = input()
temp += '%'
data += temp
print("Enter No. of Arrears:", end="")
temp = input()
temp += '%'
data += temp
print("Enter Attendance(%) :", end="")
temp = input()
temp += '%'
data += temp
print("Enter CGPA:", end="")
temp = input()
data += temp

MultiClientSocket.send(str.encode(data))
res = MultiClientSocket.recv(1024)
if Input == "I":
    print(res.decode('utf-8'))
elif Input == "D":
    print(res.decode('utf-8'))
elif Input == "U":
    print(res.decode('utf-8'))
elif Input == "C":
    print(res.decode('utf-8'))
elif Input == "V":
    x = res.decode('utf-8').split('*')
    fin = {}
    for i in range(1, len(x)):
        if (i % 2 != 0):
            temp = x[i].split("%")
            fin[x[i - 1]] = temp
    output = pd.DataFrame.from_dict(fin)
    size = len(output)
    print(output.head(size - 1))

MultiClientSocket.close()

```

OUTPUT : [SERVER-ADMINISTRATOR] – SINGLE SERVER MULTI CLIENT

Initial records.xlsx file

```
E:\Python SP>python serveradmin.py
Socket is listening on 192.168.1.36
Connected to: 192.168.1.36:63443
Total Client Connections: 1
```

Insertion

```
E:\Python SP>python admin.py
Waiting for response
I - Insert new record
U - Update a record
D - Delete a record
C - Copy all records
V - View all records
Choose from the above options: I
Enter Roll Number:CB.EN.U4ECE19001
Enter Date of Birth:18-Jun-2000
Enter Student Name:John Doe
Enter Gender:Male
Enter Branch:ECE
Enter Joining Year:2019
Enter No. of Arrears:1
Enter Attendance(%):78.93
Enter CGPA:7.82
Inserted Successfully
I - Insert new record
U - Update a record
D - Delete a record
C - Copy all records
V - View all records
Choose from the above options:
```

Updated records.xlsx file

```
E:\Python SP>python serveradmin.py
Socket is listening on 192.168.1.36
Connected to: 192.168.1.36:63443
Total Client Connections: 1
Connected to: 192.168.1.36:63514
Total Client Connections: 2
```

Update and View

```

I - Insert new record
U - Update a record
D - Delete a record
C - Copy all records
V - View all records
Choose from the above options: U
Enter Roll Number:CB.EN.U4CSE19359
Enter Date of Birth:23-Apr-2001
Enter Student Name:Neelesh Gupta
Enter Gender:Male
Enter Branch:CSE
Enter Joining Year:2019
Enter No. of Arrears:0
Enter Attendance(%):78.23
Enter CGPA:8.12
Updated Successfully
I - Insert new record
U - Update a record
D - Delete a record
C - Copy all records
V - View all records
Choose from the above options: V
      Roll Number Date of Birth      Student Name Gender Branch Joining Year No of Arrears Attendance(%) CGPA
0 CB.EN.U4CSE19346 22-Feb-2001      Raghul K B   Male   CSE      2019          0       96.0    8.35
1 CB.EN.U4CSE19358 27-Aug-2002      Devakumar     Male   CSE      2019          0       91.0    8.45
2 CB.EN.U4CSE19359 23-Apr-2001      Neelesh Gupta  Male   CSE      2019          0       78.23   8.12
3 CB.EN.U4CSE19360 15-Jul-2001      Nithin Krishna V  Male   CSE      2019          0       95.0    8.65
4 CB.EN.U4ECE19001 18-Jun-2000      John Doe     Male   ECE      2019          1       78.93   7.82
I - Insert new record
U - Update a record
D - Delete a record
C - Copy all records
V - View all records
Choose from the above options:

```

Updated records.xlsx file

Deletion

```
I - Insert new record
U - Update a record
D - Delete a record
C - Copy all records
V - View all records
Choose from the above options: D
Enter Roll Number:CB.EN.U4ECE19001
Deleted Successfully
I - Insert new record
U - Update a record
D - Delete a record
C - Copy all records
V - View all records
Choose from the above options:
```

Updated records.xlsx file

Copy and save it in another file

```
I - Insert new record
U - Update a record
D - Delete a record
C - Copy all records
V - View all records

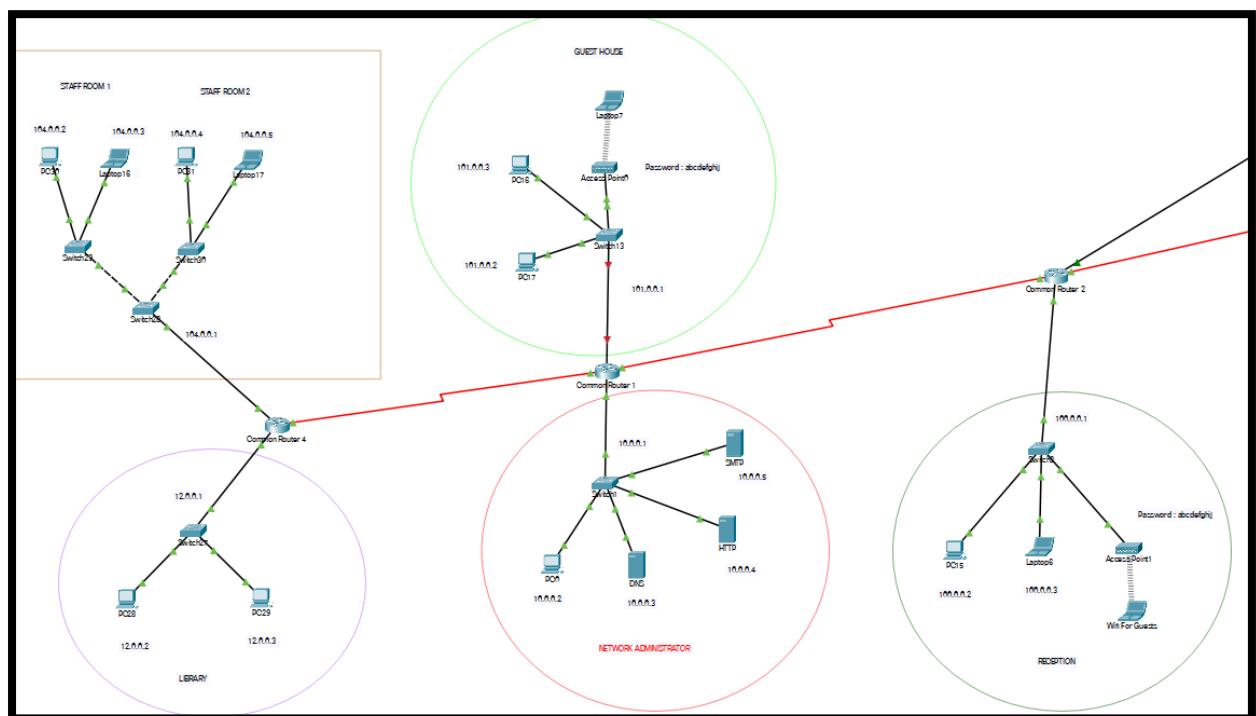
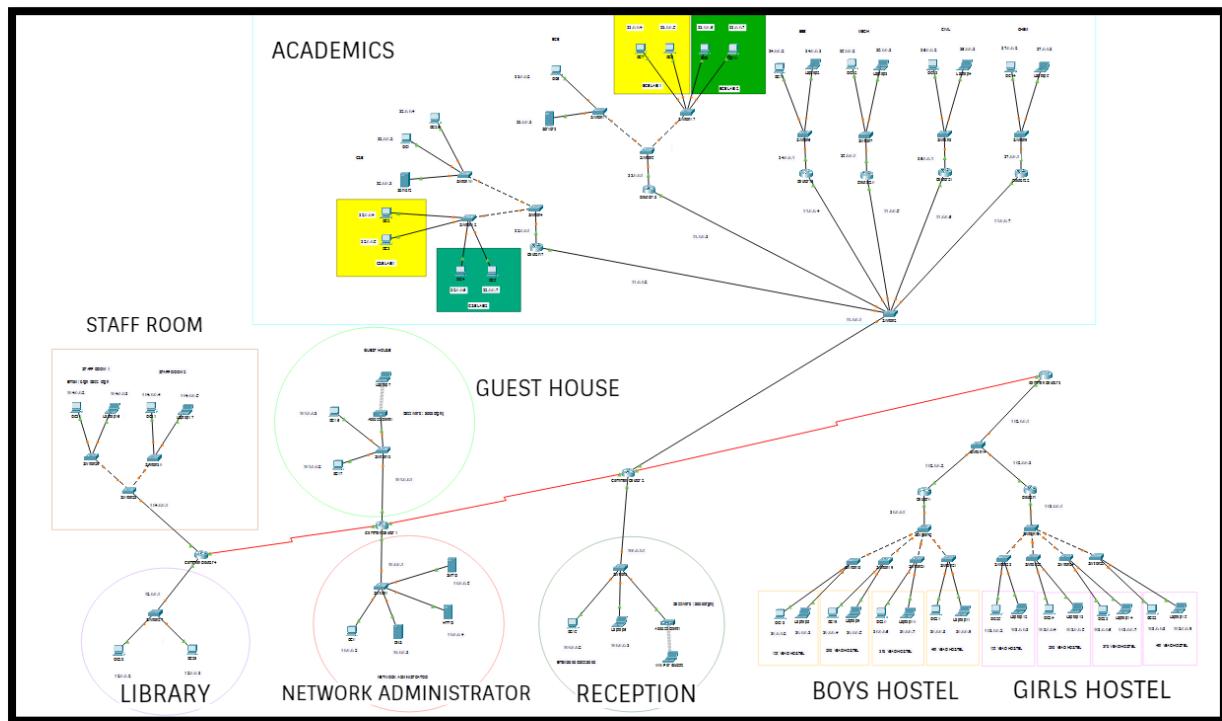
Choose from the above options: C
Copied Successfully

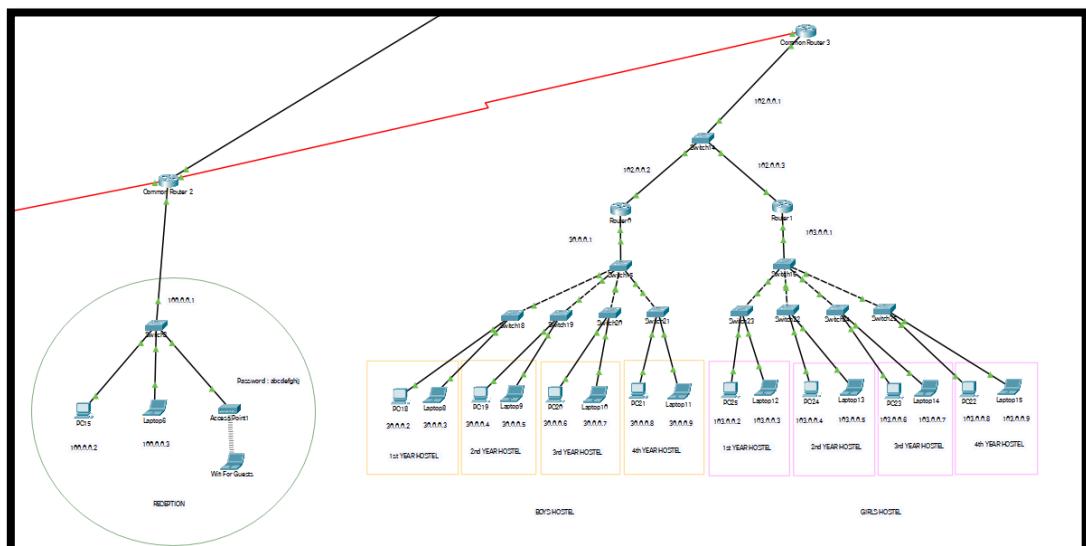
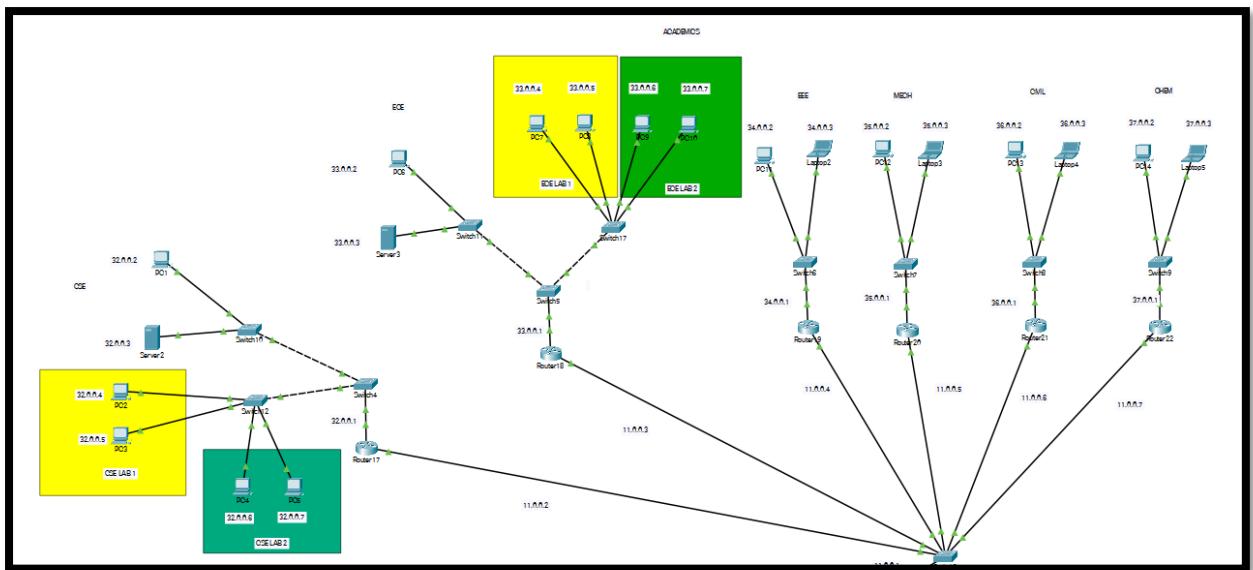
I - Insert new record
U - Update a record
D - Delete a record
C - Copy all records
V - View all records

Choose from the above options:
```

New records copy.xlsx file

CISCO PACKET TRACER DESIGN AND SETUP:





DNS :

The screenshot displays two NetworkMiner windows side-by-side, illustrating the configuration of DNS and HTTP services.

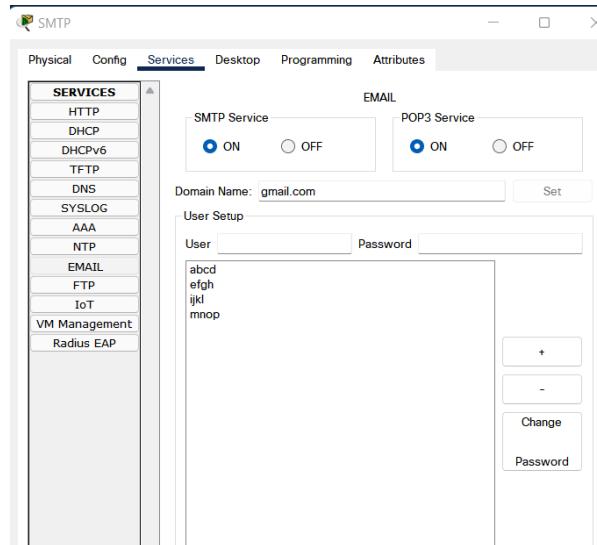
Left Window (DNS Configuration):

- SERVICES:** DNS Service is set to **On**.
- Resource Records:** A record for "dk.com" is listed with IP 10.0.0.4.

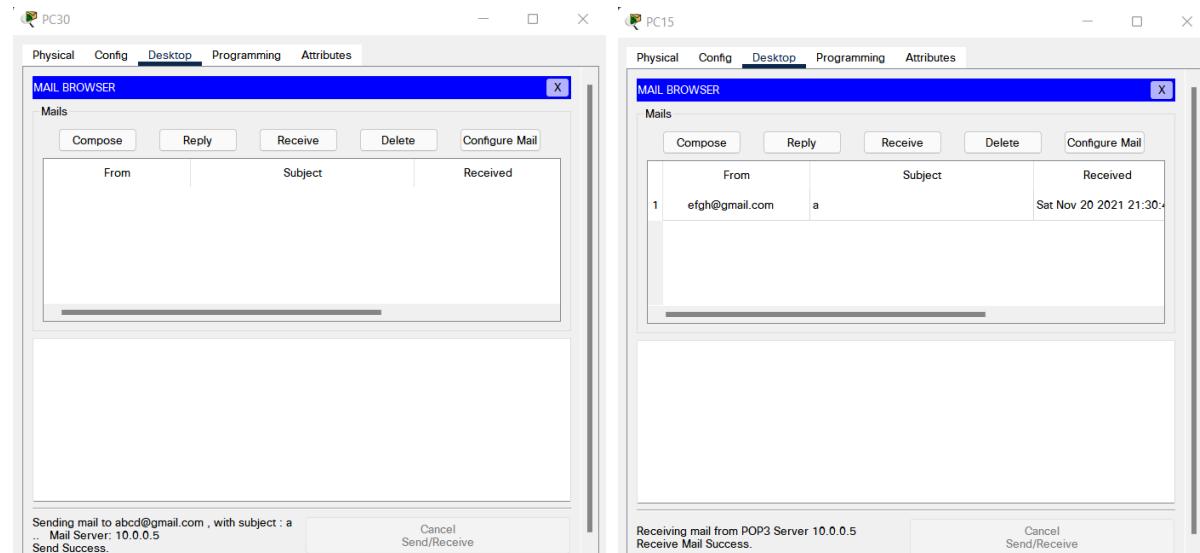
Right Window (HTTP Configuration):

- SERVICES:** Both **HTTP** and **HTTPS** services are set to **On**.
- File Manager:** A list of files including "copyrights.html", "cscoptlogo17x111.jpg", "helloworld.html", "image.html", and "index.html".

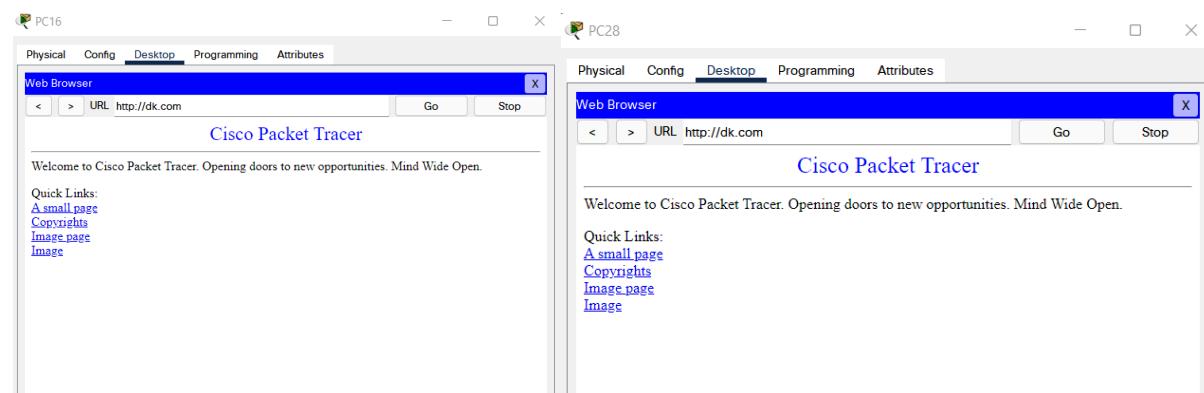
SMTP :



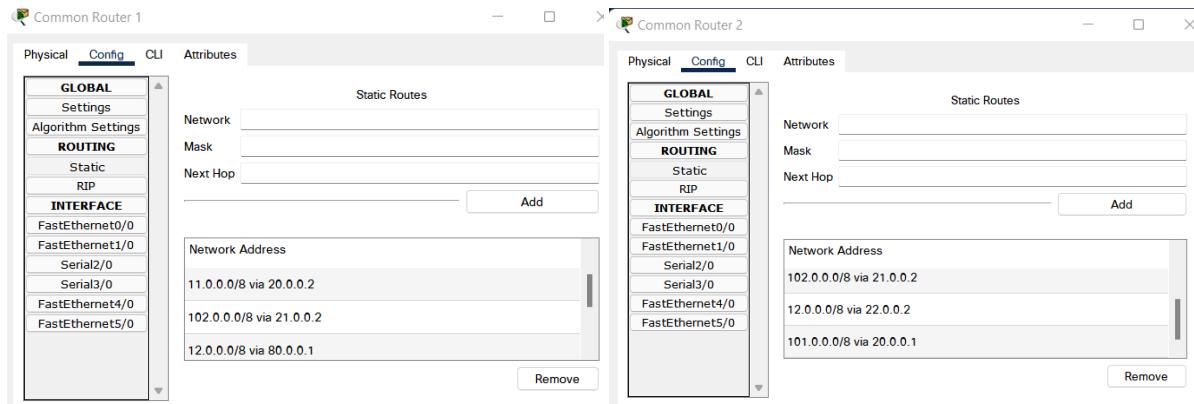
Email Service :



Pinging Servers from other PC's :



Static Routing :



Pinging other PC' via routing :

The image displays two windows of a network configuration tool, both titled "PC".

PC17 Configuration:

- Desktop** tab selected.
- Command Prompt** window:

```
Packet Tracer PC Command Line 1.0
C:\>ping 12.0.0.3

Pinging 12.0.0.3 with 32 bytes of data:

Reply from 12.0.0.3: bytes=32 time=1ms TTL=126
Reply from 12.0.0.3: bytes=32 time=1ms TTL=126
Reply from 12.0.0.3: bytes=32 time=10ms TTL=126
Reply from 12.0.0.3: bytes=32 time=1ms TTL=126

Ping statistics for 12.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 10ms, Average = 3ms
```

PC30 Configuration:

- Desktop** tab selected.
- Command Prompt** window:

```
Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=2ms TTL=126
Reply from 10.0.0.2: bytes=32 time=3ms TTL=126
Reply from 10.0.0.2: bytes=32 time=18ms TTL=126
Reply from 10.0.0.2: bytes=32 time=10ms TTL=126

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 18ms, Average = 8ms
```

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
Successful	Laptop...	PC21	ICMP	Green	0.000	N	0	(edit)	(delete)	
Successful	PC21	PC25	ICMP	Yellow	0.000	N	1	(edit)	(delete)	
Successful	PC24	Laptop13	ICMP	Magenta	0.000	N	2	(edit)	(delete)	

VLAN Configuration for Lab 1 and Lab 2 :

The screenshot shows the Cisco IOS Command Line Interface (CLI) for a switch named 'Switch12'. The 'CLI' tab is selected. The output displays system logs and a detailed VLAN configuration:

```

LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/4, changed state to up
LINK-5-CHANGED: Interface FastEthernet0/5, changed state to up
LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/5, changed state to up

Switch>show vlan brief

VLAN Name          Status    Ports
---- -----
1    default        active   Fa0/1, Fa0/6, Fa0/7, Fa0/8
                           Fa0/9, Fa0/10, 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   lab1           active   Fa0/2, Fa0/3
20   lab2           active   Fa0/4, Fa0/5
1002 fddi-default
1003 token-ring-default
1004 fddinet-default
1005 trnet-default
Switch>

```

Ctrl+F6 to exit CLI focus

Pinging VLAN to other PC and VLAN to VLAN:

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
Failed	PC1	PC2	ICMP	Purple	0.000	N	0	(edit)	(delete)	
Successful	PC2	PC3	ICMP	Blue	0.000	N	1	(edit)	(delete)	

FTP for Academic Servers :

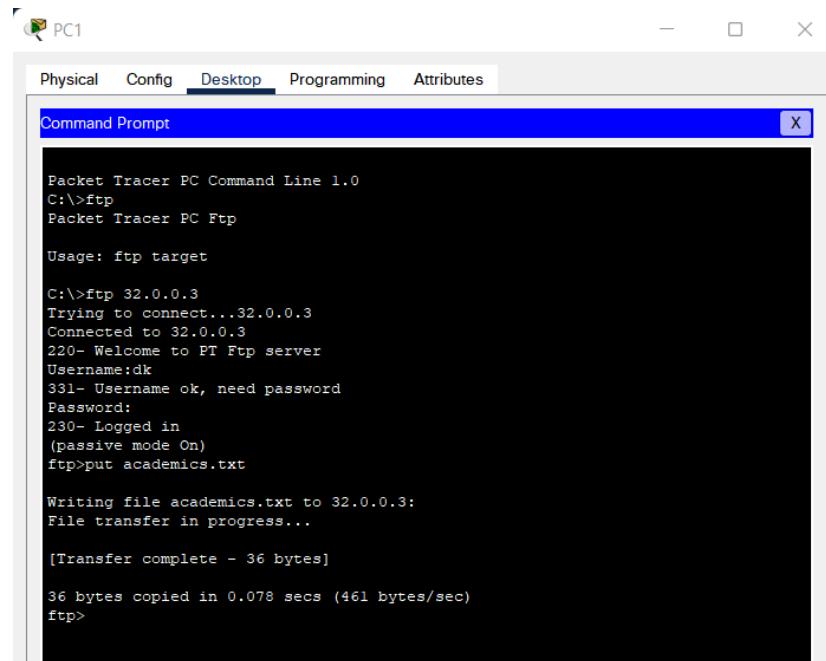
The screenshot shows the Cisco Server2 configuration interface. The 'Services' tab is selected. On the left, a sidebar lists various services: HTTP, DHCP, DHCPv6, TFTP, DNS, SYSLOG, AAA, NTP, EMAIL, FTP, IoT, VM Management, and Radius EAP. The main panel shows the 'FTP' configuration:

FTP

User Setup		
Username	Password	Permission
1 cisco	cisco	RWDNL
2 dk	dk	RWDNL

Buttons for Add, Save, and Remove are visible.

FTP Functions from PC's connected to the server :



PC1

Physical Config Desktop Programming Attributes

Command Prompt X

```
Packet Tracer PC Command Line 1.0
C:\>ftp
Packet Tracer PC Ftp

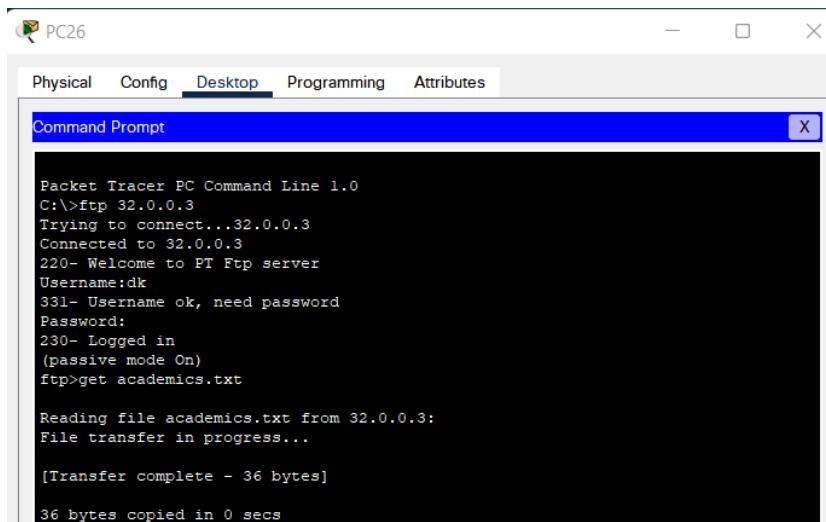
Usage: ftp target

C:>ftp 32.0.0.3
Trying to connect...32.0.0.3
Connected to 32.0.0.3
220- Welcome to PT Ftp server
Username:dk
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>put academics.txt

Writing file academics.txt to 32.0.0.3:
File transfer in progress...

[Transfer complete - 36 bytes]

36 bytes copied in 0.078 secs (461 bytes/sec)
ftp>
```



PC26

Physical Config Desktop Programming Attributes

Command Prompt X

```
Packet Tracer PC Command Line 1.0
C:>ftp 32.0.0.3
Trying to connect...32.0.0.3
Connected to 32.0.0.3
220- Welcome to PT Ftp server
Username:dk
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>get academics.txt

Reading file academics.txt from 32.0.0.3:
File transfer in progress...

[Transfer complete - 36 bytes]

36 bytes copied in 0 secs
```