AISSMS

COLLEGE OF ENGINEERING

Approved by AICTE, New Delhi, Recognized by
Govt. of Maharashtra, Affiliated to Savitribai Phule Pune University
and recognized 2(f) and 12(B) by UGC (Id.No. PU / PN/ Engg. / 093 (1992)
Accredited by NAAC with 'A+' Grade

DEPARTMENT OF COMPUTER ENGINEERING

YEAR 2021-22

TERM - 5

PRACTICAL ASSIGNMENT

CNS LABORATORY

CLASS – TE COMPUTER ENGG.(SHIFT A)
NAME – PRATIK PINGALE
ROLL NO. – 19C0056
BATCH – C

FACULTY – ANIL KADAM

Pratik Pingale 19C00056

Assignment - 1

Title:

Setup a wired LAN using Layer 2 switch and then IP switch of minimum four computers. It includes preparation of cable using line tester, configuration machine using IP addresses, testing PING utility & demonstrate the PING utility & demonstrate the PING packets captured Wireshark Packet Analyzer tool.

Outcomes:

Configuration of LAN.

Requirements:

- Hardware: Crimper, Line Tester, Cat 5, RJ45
- Software : Wireshark.

Theory:

Types of Networks:

- Common examples of area network types are:

- LAN : Local Area Network
- WLAN: Wireless Local Area Network.
- WAN: Wide Area Network
- MAN: Metropolitan Area Network
- SAN: System Area Network
- CAN: Campus Area Network

Scanned with CamScanner

- PAN : Personal Area Network
- DAN: Desk Area Network

- LAN :
  - A LAN connects network devices over a relatively short distance.
  - A networked office building, school, or home usually contains LAN.

- WAN :
  - As the term implies, a WAN spans a large physical distance.
  - The internet is the largest WAN, spanning on earth.

- MAN :
  - A network spanning a physical area larger than a LAN but smaller than WAN.
  - MAN is typically owned and operated by a single entity such as a corporation.

- Types of Cables :

  - Cable is the medium through Information usually moves from one network device to another.
  - There are severely types of cables with are commonly used with LANs.

Scanned with CamScanner

Types of cables used of cables used in networks & other related topics :

- Unshielded Twisted Pair (UTP) Cable
- Shielded Twisted Pair (STP) Cable
- Coaxial Cable
- Fiber Optic Cable.
- Cable installation guides
- Wireless LAN's

- UTP Cable :
  - Twisted pair cabling comes in following varieties:
    i) Shielded
    ii) Unshielded
  - Unshielded twisted pair (UTP) is the most populer.

- STP Cable:
  - UTP is the least expensive cable but it is susceptible to radio & electrical frequency interference, so you must place cable in environments wit lots of potential interference.
  - Shielded cables can also help to extend the maximum distance of the cables
  - Types are:
    i) indivisually shielded with foil.
    ii) Foil or braid shield inside the jacket covering all wires.
    iii) Shield around each indivisual pair.

Scanned with CamScanner

- Coaxial Cable:

  - Coaxial cabling has a single copper conductor at its center.
  - Although difficult to install it is highly resistant to signal interference.
  - Types:
    i) thick
    ii) thin

- Fiber Optic Cable:

  - Consists of glass center glass core surrounded by several layers of protective materials.
  - Ideal for large amount of electrical interference.
  - Types:
    i) single mode
    ii) multi mode

- Network Devices:

- Gateway:

  - A device sitting at a network node for interfacing with another network that uses different Protocols

- Router:

  - A router is a computer networking device that forwards data packets across a network toward their destinations, through a process called routing.

Scanned with CamScanner

- Bridge:

  - A network bridge connects multiple network segments at the data link layer.

- Switch :

  - A switch is a device that allocates traffic from one network segment to certain lines which connect the segment to another network segment.

- Hub:

  - An ethernet hub or concentrator connects multiple ethernet segments together making them act as a single segment.

- Repeater:

  - A repeater is an electronic device that receives weak or low level signal & retransmits it at a higher level or power, so that the signal can cover longer distances without degradation.

- Procedure:

- There are three main types of networking cables:
  i) straight through
  ii) crossover
  iii) rollover

each cable has distinct use.

Scanned with CamScanner

- Use a straight through cable when :

  - connecting router to a hub.
  - connecting computer to a switch.
  - Connecting to a LAN port to a switch, hub or computer.

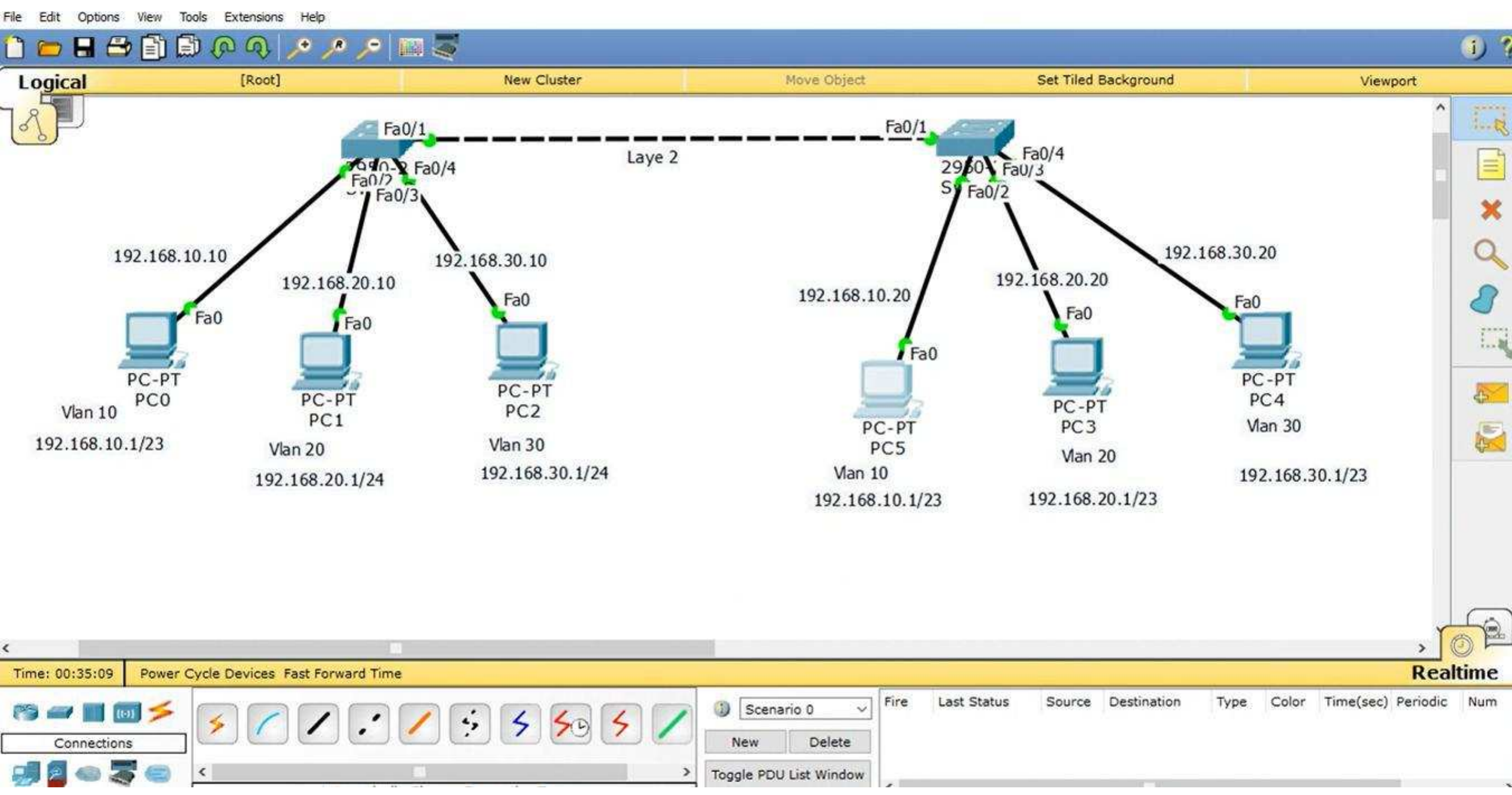
- Use a crossover cable when :

  - connecting computer to router.
  - connecting computer to computer.
  - Connecting router to router.
  - Connecting switch to switch
  - Connecting hub to hub.

- Conclusion :

  - Hence we have studied to setup a wired LAN.

Scanned with CamScanner


File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Fa0/1
Fa0/4
Fa0/2
Fa0/3
Laye 2
Fa0/1
Fa0/4
Fa0/3
Fa0/2
192.168.10.10
192.168.20.10
192.168.30.10
192.168.10.20
192.168.20.20
192.168.30.20
Fa0
PC-PT
PC0
Vlan 10
192.168.10.1/23
PC-PT
PC1
Vlan 20
192.168.20.1/24
PC-PT
PC2
Vlan 30
192.168.30.1/24
PC-PT
PC5
Vlan 10
192.168.10.1/23
PC-PT
PC3
Vlan 20
192.168.20.1/23
PC-PT
PC4
Vlan 30
192.168.30.1/23
Time: 00:35:09
Power Cycle Devices Fast Forward Time
Realtime
Connections
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num

Pratik Pingale 19C0056

Assignment-2

- Title :

Demonstrate the different types of topologies and types of transmission media by using a packet tracer tool.

- Outcomes :

Demonstration of network topologies.

- Requirement :

Software : Cisco packet tracer.

- Theory :

- Network Topologies :

- The study of network topology recognizes seven basic topologies :

  - point to point topology
  - Bus topology
  - Star topology
  - Ring topology
  - Tree topology
  - Mesh topology
  - Hybrid topology

- Classification is based on the interconnection between computers it could be :

  - physical
  - logical

Scanned with CamScanner

- The physical topology of a network is determined by the capabilities of the network access devices and media, the level control or fault tolerance desired.

- Bus:

- Bus Network Topology:

- In local are networks where bus topology is used, each machine is connected to a single cable

- Each computer or server is connected to the single bus ~~capable~~ cable through some kind of connector.

- A terminator is required at each end of the bus to prevent the signal from bouncing back and forth on the bus cable.

- Star Network Topology:

- A star topology is designed with each node connected directly to a central network hub, switch, or concentrator.

- Data on a star network passes through the hub, switch or concentrator before continuing to its destination.

- Advantages:

- Easy to install & wire

- Easy to detect faults & to remove parts.

Scanned with CamScanner

- • Tree or Expanded Star Topology:

- A tree topology combines characteristics of linear bus & star topologies.

- Tree topologies allow for the expansion of an existing network, and enable schools to configure a network to meet their needs.

• Advantages:

- point to point wiring for individual segments.

- Supported by several hardware and software vendors.

• Disadvantages:

- Overall length of each segment is limited.

- If the backbone line breaks, the entire segment goes down.

- More difficult to wire.

• Ring Topology:

- A ring topology is a network configuration where device connections create a circular data path.

- Each networked device is connected to two others, like points on a circle.

Scanned with CamScanner

- Together devices in a ring topology are referred to as a ring network.

• Disadvantages:

- If any indivisual connection in the ring is broken, the entire network is affected.

• Mesh Topology:

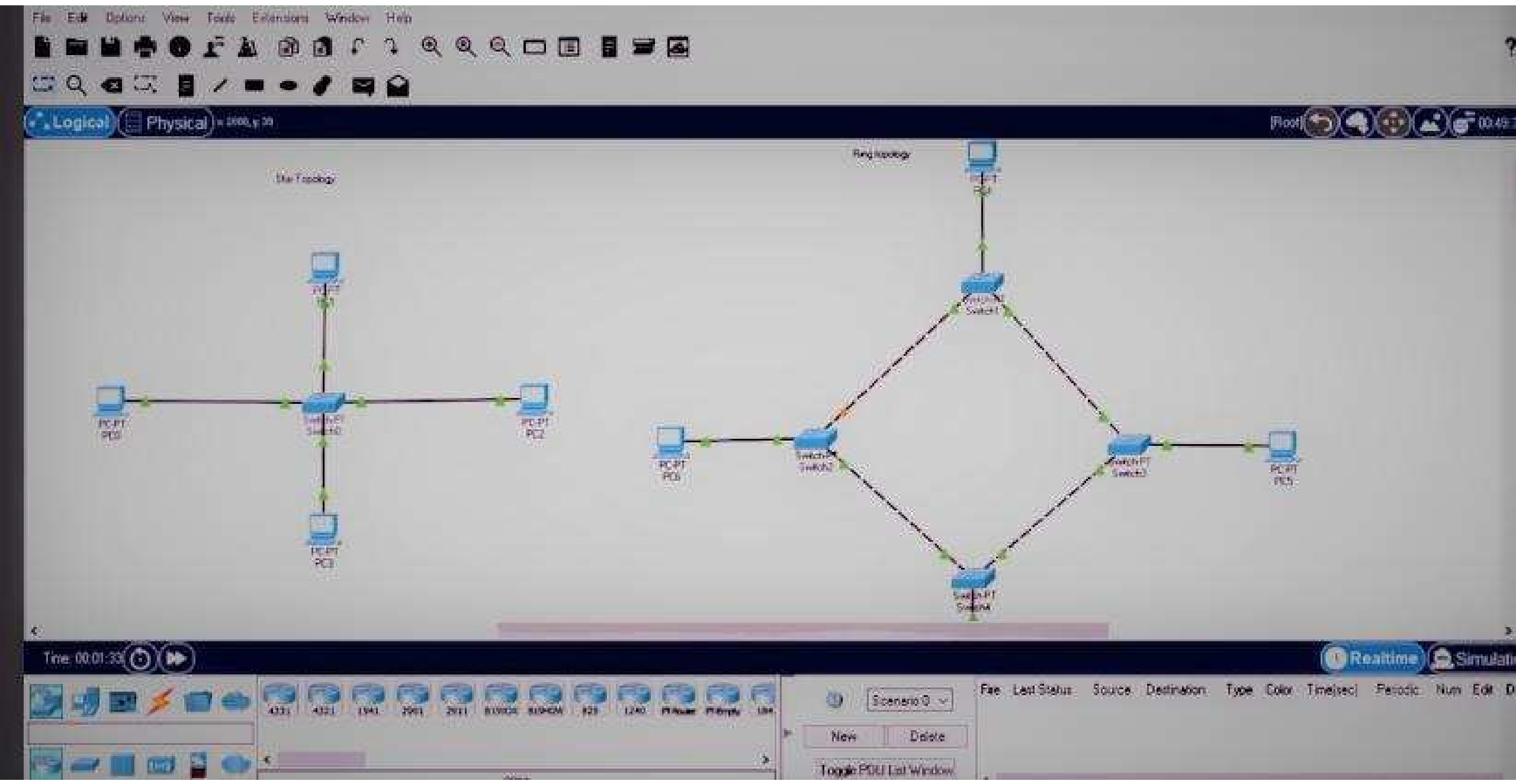
- A mesh topology is a network setup where each computer & device its interconnected.

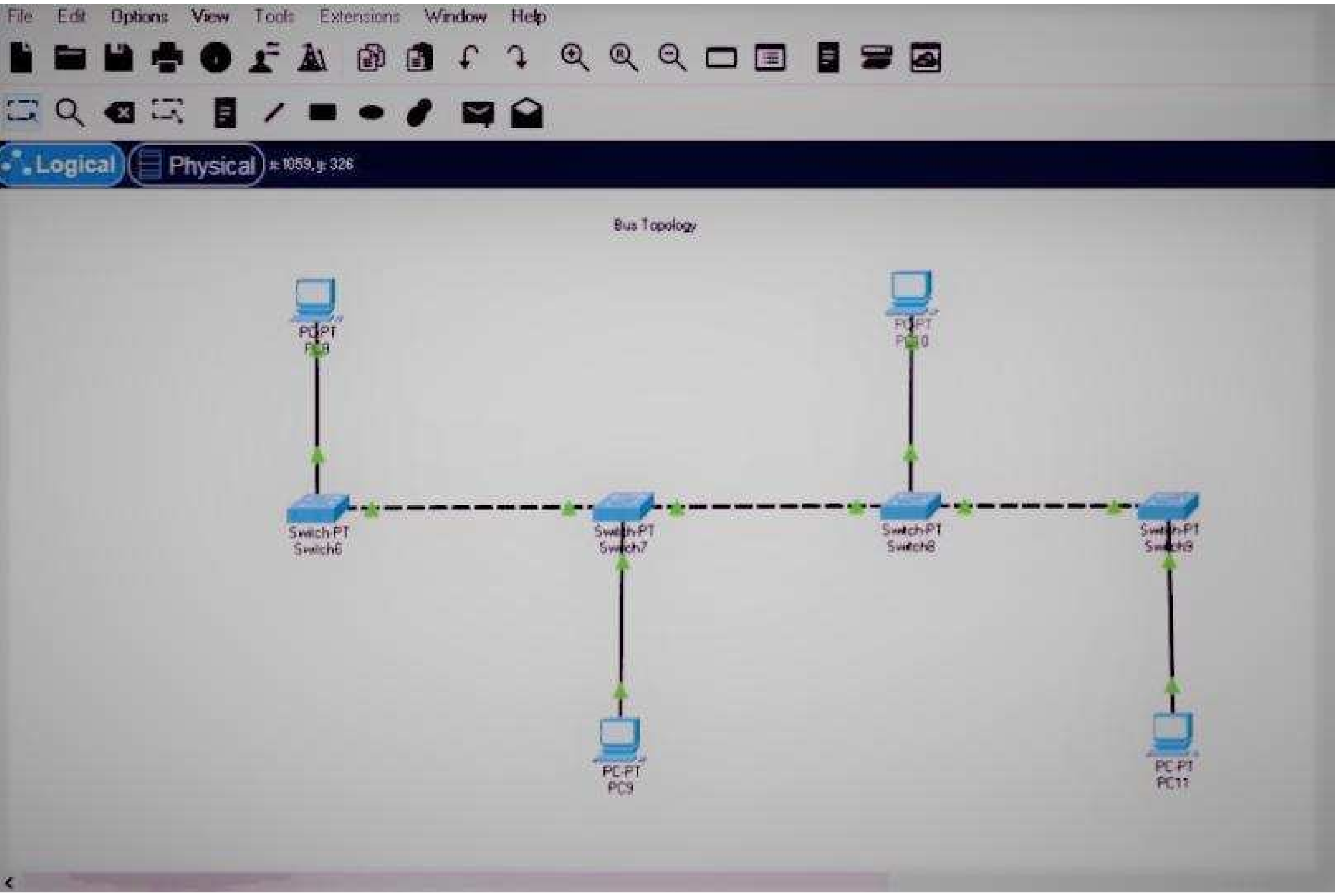
- This topology setup allows for most transmissions to be distributed even if one of the connections goes down.

• Conclusion:

- Hence, we have studied to implement the difficult network topologies on CISCO packet Tracer.

Scanned with CamScanner


File Edit Options View Tools Extensions Window Help
Logical
Physical
Ring topology
Star Topology
Realtime
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num Edit


File Edit Options View Tools Extensions Window Help
Logical
Physical
Bus Topology
PC-PT
Switch-PT
Switch6
Switch-PT
Switch7
Switch-PT
Switch8
PC-PT
PC9
PC-PT
PC11

Pratik Pingale 19C00056

Assignment-3

- Title:

Write a program for error detection & correction for 7/8 bits ASCII codes using Hamming Codes using Hamming Codes or CRC.

- Outcomes:

Demonstrate CRC with example.

- Theory:

Cyclic Redundancy check: CRC

- CRC works with a simple technique.

- Whenever the data is to be transmitted, a computed amount of check bits are appended to the data.

- These bits are also called checksum.

- Once it is assigned to the data, the data is then transited to the destination.

- The checksum is also maintained at the destination side.

- Upon receiving the data in the destination, it is checked for all the errors using those checksums.

- The goal is to ensure using those checksum of the received data is same as the checksum received.

Scanned with CamScanner

- If there are any mathematical data changes, a negative negative acknowledgement is sent.

- Hence, the data is re-transmitted and hence the user receives the correct data transmission.

- Example :

```
                    Qoutient
Divisor             111101
  1101   )  100100000
             1101
              1000
              1101
               1010
               1101
                0101 0
                1101
                 1110
                 1101
                  0110
                  0000
                   1100
                   1101
Remainder →         001
```

- Conclusion :

- Hence, we have learned the techniques for error detection & correct.

Scanned with CamScanner

5. Code for 7/8 bit CRC & Hamming Code.

```
#include
<iostream> using
namespace std; int
main() {
   int data[10];
   int dataatrec[10], c, c1, c2, c3, i;

   cout << "Enter 4 bits of data one by
   one\n"; cin >> data[7];
   cin >>
   data[6]; cin
   >> data[5];
   cin >>
   data[3];

   data[4] = data[5] ^ data[6] ^
   data[7]; data[2] = data[3] ^
   data[6] ^ data[7]; data[1] =
   data[3] ^ data[5] ^ data[7];

   cout << "\nEncoded data
   is\n"; for (i=1;i<=7;i++) {
      cout << data[i];

   }

   cout << "\n\nEnter received data bits one by
   one\n"; for (i=1;i<=7;i++) {
      cin >> dataatrec[i];
   }

   c1 = dataatrec[1] ^ dataatrec[3] ^ dataatrec[5] ^
   dataatrec[7]; c2 = dataatrec[2] ^ dataatrec[3] ^
   dataatrec[6] ^ dataatrec[7]; c3 = dataatrec[4] ^
   dataatrec[5] ^ dataatrec[6] ^ dataatrec[7]; c =
   c3*4+c2*2+c1;

   if (c == 0) {

      cout << "\ncongratulations there is no error: ";

   }

   else {

      cout << "\nerror on the postion: " <<
      c; cout << "\nCorrect message is: ";
      if (dataatrec[c]==0)
```

```
    { dataatrec[c]=1;
  }
  else {
    dataatrec[c]=0;
  }
  for (i=1;i<=7;i++) {
    cout <<
    dataatrec[i];
  }
    }
    return 0;
}
```

Output:

```
PS C:\Users\Gaurav\Documents\VS Code>  & 'c:\Us
t-kvy2mhhz.2fl' '--stderr=Microsoft-MIEngine-Er
n\gdb.exe' '--interpreter=mi'
Enter 4 bits of data one by one
1
0
0
0
1

Encoded data is
0011001

Enter received data bits one by one
0
0
0
1
0
0
1

error on the postion: 3
Correct message is: 0011001
PS C:\Users\Gaurav\Documents\VS Code>
```

```
Enter 4 bits of data one by one
1
3
5
7


Encoded data is
1357753


Enter received data bits one by one
1
2
5
4
7
5
7


Error on position 14
Data sent : 1357753
Data received : 1254757
Correct message is
1


...Program finished with exit code 0
Press ENTER to exit console.
```

```
Enter 4 bits of data one by one

1
2
3
4

Encoded data is
1230476

Enter received data bits one by one
1
2
3
0
4
7
6

No error while transmission of data

...Program finished with exit code 0
Press ENTER to exit console.
```

```
Enter Data: dataset
Enter Key: qwer

Quotient is: d111111
Remainder is: 111
Final data is: dataset111

...Program finished with exit code 0
Press ENTER to exit console.
```

Pratik Pingale 19C0056

Assignment-4

- Title:

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

- Outcomes:

Implementation of Flow Control Mechanism in network.

- Theory:

- The basic idea of sliding window protocol is that both sender and receiver keep a "window" of acknowledgement.

- Sender keeps the value of expected acknowledgement.

- The receiver keeps the value of expected receiving frame.

- When it receives an acknowledgement from the receiver, the sender advances the window.

- When it receives the expected frame, the receiver advances advances the window.

- The flow of data should not be allowed to overwhelm the receiver.

Scanned with CamScanner

- Sliding Window Protocol:

  - With the use of multiple frames for a single message, the stop & wait protocol does not perform well.

  - Only one frame at a time can be in transmit.

  - Efficiency can be greatly improved by allowing multiple frames to be in transit at the same time & making use of fullduplex line

  - To keep tracks of frames, sender station sends sequencially numbered frames.

  - If the header of the frames allows k bits, the sequence numbers range should be from 0 to 2k-1

  - The size of sender's window is at most 2k-1.

- Go back N ARQ:

  - In the Go back N protocol, the sequence numbers are modulo 1|'', where size of sequence number field in bits.

- Selective Repeat ARQ:

  - Go back N ARQ simplifies the process at the receiver site

  - The receiver keeps track of only one variable.

  - There is no need to buffer out of order frames.

Scanned with CamScanner

- Key differences between Go back N & Selective Repeat:

i) Go back N protocol is design to retransmit all the frames that are arrived after the damaged or a lost frame. Selective Repeat protocol retransmits only that is damaged or lost.

ii) Go back N has a window size of N-1 & selective repeat have a window size <= (N+1)/2

iii) Neither sender nor receiver need the sorting algorithm in Go-Back-N whereas, receiver must be able to sort the as it as it has to maintain the sequence.

iv) Generally, the Go-back-N is more is use due to its less complex nature instead of Selective Repeat Protocol.

- Conclusion:

-Thus, we have studied & implemented Sliding Window Protocol.

Scanned with CamScanner

Administrator: Command Prompt

```
Localhost/127.0.0.1
.......Client........
Connect
Enter the number of frames to be requested to the server
5
Enter the type of trans. Error=1 ; No Error=0
1
received frame no: 0
Sending acknowledgement for frame no: 0
received frame no: 1
Sending acknowledgement for frame no: 1
Discarded frame no: 3
Sending NEGATIVE ack
Discarded frame no: 4
Sending NEGATIVE ack
received frame no: 2
Sending acknowledgement for frame no: 2
received frame no: 3
Sending acknowledgement for frame no: 3
received frame no: 4
Sending acknowledgement for frame no: 4
Quiting

C:\Users\Rajendra\Desktop\CNS Practical\SlidingWindow(GObackN)>
```

Administrator: Command Prompt

```
Localhost/127.0.0.1
.......Client........
Connect
Enter the number of frames to be requested to the server
5
Enter the type of trans. Error=1 ; No Error=0
0
received frame no: 0
Sending acknowledgement for frame no: 0
received frame no: 1
Sending acknowledgement for frame no: 1
received frame no: 2
Sending acknowledgement for frame no: 2
received frame no: 3
Sending acknowledgement for frame no: 3
received frame no: 4
Sending acknowledgement for frame no: 4
Quiting
```

Administrator: Command Prompt

```
                    should always be passed as the argument to the -splash option. 
                    The most appropriate scaled image provided will be picked up
                    automatically.
                    See the SplashScreen API documentation for more information
  @argument files
                    one or more argument files containing options
  -disable-@files
                    prevent further argument file expansion
  --enable-preview
                    allow classes to depend on preview features of this release
o specify an argument for a long option, you can use --<name>=<value> or
-<name> <value>.

:\WINDOWS\system32>cd C:\Users\Rajendra\Desktop\CNS Practical\SelectiveRepeat

:\Users\Rajendra\Desktop\CNS Practical\SelectiveRepeat>java server
rror: Could not find or load main class server
aused by: java.lang.ClassNotFoundException: server

:\Users\Rajendra\Desktop\CNS Practical\SelectiveRepeat>java Server
rror: Could not find or load main class Server
aused by: java.lang.ClassNotFoundException: Server

:\Users\Rajendra\Desktop\CNS Practical\SelectiveRepeat>java Server.java
aiting for connection
he number of packets sent is:8

:\Users\Rajendra\Desktop\CNS Practical\SelectiveRepeat>
```

Administrator: Command Prompt

```
ava
...........Server..........
Waiting for connection....
Received request for sending frames
Sending....
sending frame no 0
Waiting for acknowledgement
received ack for frame no: 0 as 0
sending frame no 1
Waiting for acknowledgement
received ack for frame no: 1 as 1
sending frame no 2
sending frame no 3
Waiting for acknowledgement
sending frame no 4
Waiting for acknowledgement
Resending frame 2
Waiting for ack
received ack for frame no: 2 as 2
Resending frame 3
Waiting for ack
received ack for frame no: 3 as 3
Resending frame 4
Waiting for ack
received ack for frame no: 4 as 4
Quiting
```

Administrator: Command Prompt

```
.............Server...........
Waiting for connection....
Received request for sending frames
Sending....
sending frame number 0
Waiting for acknowledgement
received acknowledgement for frame 0 as 0
sending frame number 1
Waiting for acknowledgement
received acknowledgement for frame 1 as 1
sending frame number 2
Waiting for acknowledgement
received acknowledgement for frame 2 as 2
sending frame number 3
Waiting for acknowledgement
received acknowledgement for frame 3 as 3
sending frame number 4
Waiting for acknowledgement
received acknowledgement for frame 4 as 4
Quiting
```

Pratik Pingale 19C00056

Assignment-5

Title:

Write a program to demonstrate subnetting & find the subnetting masks.

Objective:

To understand working of subnetting and find the subnet masks.

Theory:

- Internet Protocol being a layer 3 protocol (OSI) takes data from layer-4 (Transport) and divides it into packets.

- IP packet encapsulates data unit received from above layer and add to its own header information.

- The encapsulated data is referred to as IP payload.

- Unicast Addressing Mode:

  - data is sent only to one destined host.

  - here the client sends data to the targeted server.

- Broadcast Addressing Mode:

  - the packet is addressed to all the host in a network segment.

  - here the client sends a packet, which is entertained by all servers.

Scanned with CamScanner

- Hierarchical Addressing Scheme :

  - IPv4 uses hierarchical addressing scheme.

  - An IP address, which is 32-bits in length, is divided into two or three parts.

- Subnet Mask :

  - 32 bit IP address contains information about host and its network

  - It is very necessary to distinguish both for this routers use subnet mask.

- Binary Representation :

  - Simplest form of converting binary from decimal value.

  - IP address is 32-bit value which is divided into 4 octets.

  - A binary octet contains 8 bits and the value of each bit can be determined by the position of bit value '1' in the octet.

- Class A address :

  - The first bit of the first octet is always set to 0.

  - Thus the first octet ranges from 1-127, i.e. Class A addresses only include IP starting from 1.x.x.x to 126.x.x.x only

Scanned with CamScanner

- Class B address:

  - An IP address which belongs to class B has the first two bits in the first octet set to 10.

  - i.e. Class B IP Addresses range from 128.0.x.x to 191.255.x.x.

- Class C address:

  - The first octet of class C IP address has its first 3 bits set to 110

  - C

  - Class C IP addresses range from 192.0.0.x to 223.255.255.x.

- Class D address:

  - Very first four bits of the first octet in class D IP addresses are set to 1110.

  - Class D IP address ranges from 224.0.0.0 to 239.255.255.255.

- Class E address:

  - This IP class is reserved for experimental purposes only for R & D or study.

  - IP addresses ranges from 240.0.0.0 to 255.255.255.254

- Class A subnets:

  - In class A, only the first octet is used as network identifier & rest of three octets are used to be assigned to Hosts.

Scanned with CamScanner

- Class B subnets:

  - Class B IP Addresses can be subnetted ~~wi~~ the same way as class A addresses, by borrowing bits from host bits.

- Class C subnets:

  - Class C IP addresses are normally assigned to a very small network because it can only have 254 hosts in a network.

- Conclusion:

  Thus, we have studied working of subletting and find the subnet masks.

Scanned with CamScanner

```
Enter the ip address: 192.168.1.1
IP in binary is 11000000101010000000000100000001
Enter the number of addresses: 2
Number of bits required for address = 1
The subnet mask is = 31
Subnet address is = 192.168.1.0
Broadcast address is = 192.168.1.1


...Program finished with exit code 0
Press ENTER to exit console.
```

Pratik Pingale 19C0056

Assignment-6

Title:

Write a program to implement link state / Distance vector routing protocol to find suitable path for transmission.

Objectives:

To understand working of distance vector routing protocol.

Theory:

- Routing Algorithm:

  - The process which is used to establish the routes by following the data packets to reach the destination.

  - This kind of algorithm mainly determines the best lane to take mathematically.

- Adaptive Algorithm:

  - Used to change the decisions of routing when traffic load & network topology changes.

  - These changes will be reflected within the topology & network traffic.

- Non Adaptive Algorithm:

  - Do not modify their routing decisions.

  - Also called static routing.

Scanned with CamScanner

- Distance vector routing:

i) Distributed:

-It is distributed in that each node is receiving information from one or more of its neighbours, performs calculations & then distributes the result.

ii) Iterative:

- It is iterative in that its process continues until no more information is available to be exchanged.

iii) Asynchronous:

- It does not require that all of its nodes operate in the lock step with each other.

- Algorithm:

1 Start

2 By convention, the distance node is assigned to 0 and when a node is unreachable distance is accepted as 999.

3 Accept the input distance matrix from the user.

4 Store the distance between nodes in a suitable variable

5 Calculate the minimum distance between nodes.

6 If distance is larger than calculated alternative path replace it with calculated distance

7 Print the shortest path calculated

8. stop

Scanned with CamScanner

Conclusion :

- Hence, we have ~~calculated~~ studied working of distance vector routing protocol to find suitable path for transmission.

Scanned with CamScanner

```
1-Load File
2-Build Routing Table for Each Router
3-Out Optimal Path and Minimum Cost
1
Please load original routing table data file:
case0.txt
Original routing table is as follows:
0        2        5        1        -1
2        0        8        7        9
5        8        0        -1       4
1        7        -1       0        2
-1       9        4        2        0

1-Load File
2-Build Routing Table for Each Router
3-Out Optimal Path and Minimum Cost
2
Please select a router:
1
destination: 1  next-hop: 0
destination: 2  next-hop: 2
destination: 3  next-hop: 3
destination: 4  next-hop: 4
destination: 5  next-hop: 4

1-Load File
2-Build Routing Table for Each Router
3-Out Optimal Path and Minimum Cost
3
Please input the source and destination router number:
1 5
The shortest path from 1 to 5 is 1-4-5, the total cost is 3.

1-Load File
2-Build Routing Table for Each Router
3-Out Optimal Path and Minimum Cost
```

```
1-Load File
2-Build Routing Table for Each Router
3-Out Optimal Path and Minimum Cost
1
Please load original routing table data file:
case3_star.txt
Original routing table is as follows:
0        1        1        1        1        1        1        1        1        1
1        0        -1       -1       -1       -1       -1       -1       -1       -1
1        -1       0        -1       -1       -1       -1       -1       -1       -1
1        -1       -1       0        -1       -1       -1       -1       -1       -1
1        -1       -1       -1       0        -1       -1       -1       -1       -1
1        -1       -1       -1       -1       0        -1       -1       -1       -1
1        -1       -1       -1       -1       -1       0        -1       -1       -1
1        -1       -1       -1       -1       -1       -1       0        -1       -1
1        -1       -1       -1       -1       -1       -1       -1       0        -1
1        -1       -1       -1       -1       -1       -1       -1       -1       0

1-Load File
2-Build Routing Table for Each Router
3-Out Optimal Path and Minimum Cost
2
Please select a router:
1
destination: 1  next-hop: 0
destination: 2  next-hop: 2
destination: 3  next-hop: 3
destination: 4  next-hop: 4
destination: 5  next-hop: 5
destination: 6  next-hop: 6
destination: 7  next-hop: 7
destination: 8  next-hop: 8
destination: 9  next-hop: 9
destination: 10 next-hop: 10

1-Load File
2-Build Routing Table for Each Router
3-Out Optimal Path and Minimum Cost
```

```
1-Load File
2-Build Routing Table for Each Router
3-Out Optimal Path and Minimum Cost
2
Please select a router:
2
destination: 1  next-hop: 1
destination: 2  next-hop: 0
destination: 3  next-hop: 1
destination: 4  next-hop: 1
destination: 5  next-hop: 1
destination: 6  next-hop: 1
destination: 7  next-hop: 1
destination: 8  next-hop: 1
destination: 9  next-hop: 1
destination: 10 next-hop: 1

1-Load File
2-Build Routing Table for Each Router
3-Out Optimal Path and Minimum Cost
3
Please input the source and destination router number:
2 3
The shortest path from 2 to 3 is 2-1-3, the total cost is 2.

1-Load File
2-Build Routing Table for Each Router
3-Out Optimal Path and Minimum Cost
```

Pratik Pingale 19C0056

Assignment - 7

- **Title:**

Use packet tracer tool for configuration of 3 router network using one of the following protocol RIP.

- **Outcomes:**

Configuration of RIP in CISCO Packet Tracer.

- **Theory:**

- The routing information protocol (RIP) is one of the oldest distance vector routing protocols which employ the hop count as a routing metric

- RIP prevents routing loops by implementing a limit on the number of hops allowed in a path from source to destination.

- The maximum no. of hops allowed for RIP is 15.

- A hop of count 16 is considered an infinite distance & the route is considered inreachable.

- Each RIP router transmitted full updates every 30 seconds.

- In the early deployments, routing tables were small enough that the traffic was not significant.

- As networks grew in size, however it became evident there could be a massive traffic outbreak in every 30 seconds.

Scanned with CamScanner

- Sally Floyd and Van Jacobson showed in 1994 that without slight randomisation of the update timer, the timers synchronised over time.

- RIP is not the preferred choice for routing as its time to coverage and scalability are poor compared to EIGRP, OSPF or IS-IS.

- But RIP is easy to configure.

- RIP uses the User Diagram Protocol (UDP) as its transport protocol and is assigned the reserved port no. 520.

- CISCO Packet Tracer:

- It is a cross platform visual simulation tool designated by CISCO systems.

- It allows users to create network topologies and imitate modern computer networks.

- Software allows user to simulate the configuration of CISCO routers and switches using a simulated command line interface.

- Makes use of drag & drop user interface.

- Can run on IOS, Linux and Microsoft Windows.

- Similar android app is also available.

- Allows user to create simulated network topologies.

Scanned with CamScanner

- It can also be used in collaboration.

- It also allows user (instructors) to create activities that students have to complete.

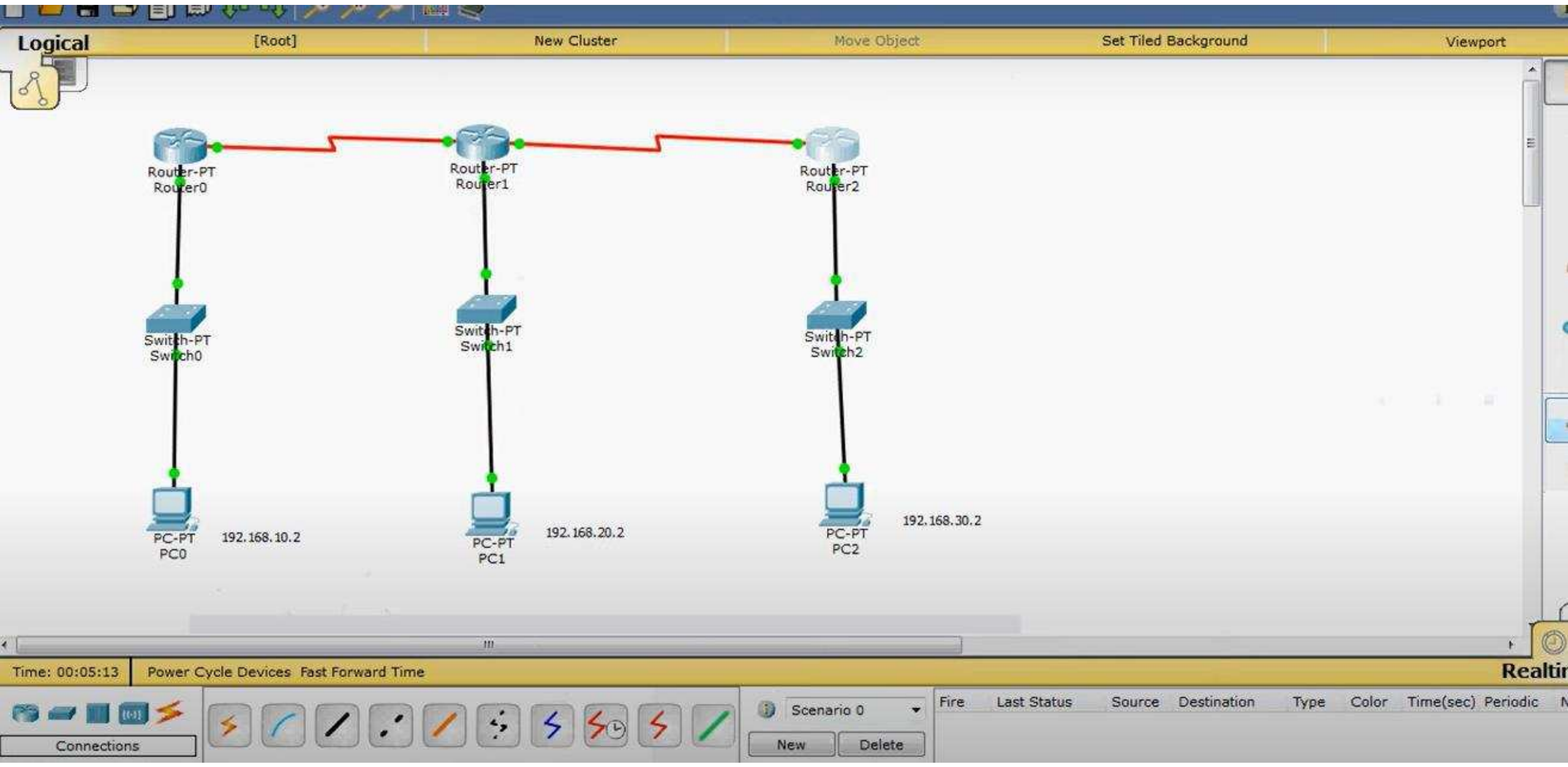
- It is often used in educational settings as a learning aid.

- Cisco claims that it is also useful for network experimentation.

- Conclusion:

We learned how to configure RIP using packet tracer.

Scanned with CamScanner


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router0
Router-PT
Router1
Router-PT
Router2
Switch-PT
Switch0
Switch-PT
Switch1
Switch-PT
Switch2
PC-PT
PC0
192.168.10.2
PC-PT
PC1
192.168.20.2
PC-PT
PC2
192.168.30.2
Time: 00:05:13
Power Cycle Devices
Fast Forward Time
Connections
Scenario 0
New
Delete
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic

Pratik Pingale 19C0056

Assignment -8

- Title:

Write a program using TCP socket for wired network for following:

a) Say Hello to Each other (for all students)

b) File trasfer (For all students)

c) Calculator (Arithmetic) (50% students)

d) Calculator (Trigonometry) (50% students)

- Requirements:

- Fedora 20 with Pentium IV and above
- 1 GB RAM
- 120 GB HDD
- Monitor
- Keyboard
- Mouse
- Modelio
- Eclipse
- CDT
- Python Interpreter
- Pydev
- J2SE
- Wireshark Packet Analyzer Tool

- Theory:

a) Say Hello to Each Other:

1. TCP Socket Programming for wired network:

- The two key classes from the java.net package used in creation of server and client programs are: Σ ServerSocket Σ Socket

Scanned with CamScanner

- A server program creates a specific type of socket that is used to listen for client requests.

- In the case of a connection request, the program creates a new socket through which it will exchange data with the data client using input & output streams.

- A simple ~~pr~~ server program in java:

1. Open the Server Socket
2. Wait for the Client Request
3. Create input streams for communicating to the client
4. Perform communication ~~or I/O with the server~~ with client receive from
5. client
5. Close socket

- A simple Client Program in java:

1. Create a Socket Object
2. Create I/O streams for communicating with the server.
3. Perform I/O or communication with the server.
4. Close the socket when done.

2. Running Socket Programs:

- Compile both server & client programs & then deploy server program which is going to act as a client.

- If required both client & server program can run on the same machine.

- The client program can run on any computer in the network, as

Scanned with CamScanner

long as there is no firewall between them.

- The client program is just establishing a connection with the server & then waits for a message.

b) File Transfer :

- A TCP client initiates the communication with a server which is waiting for the connection.

- TCP is connection oriented and UDP is connectionless

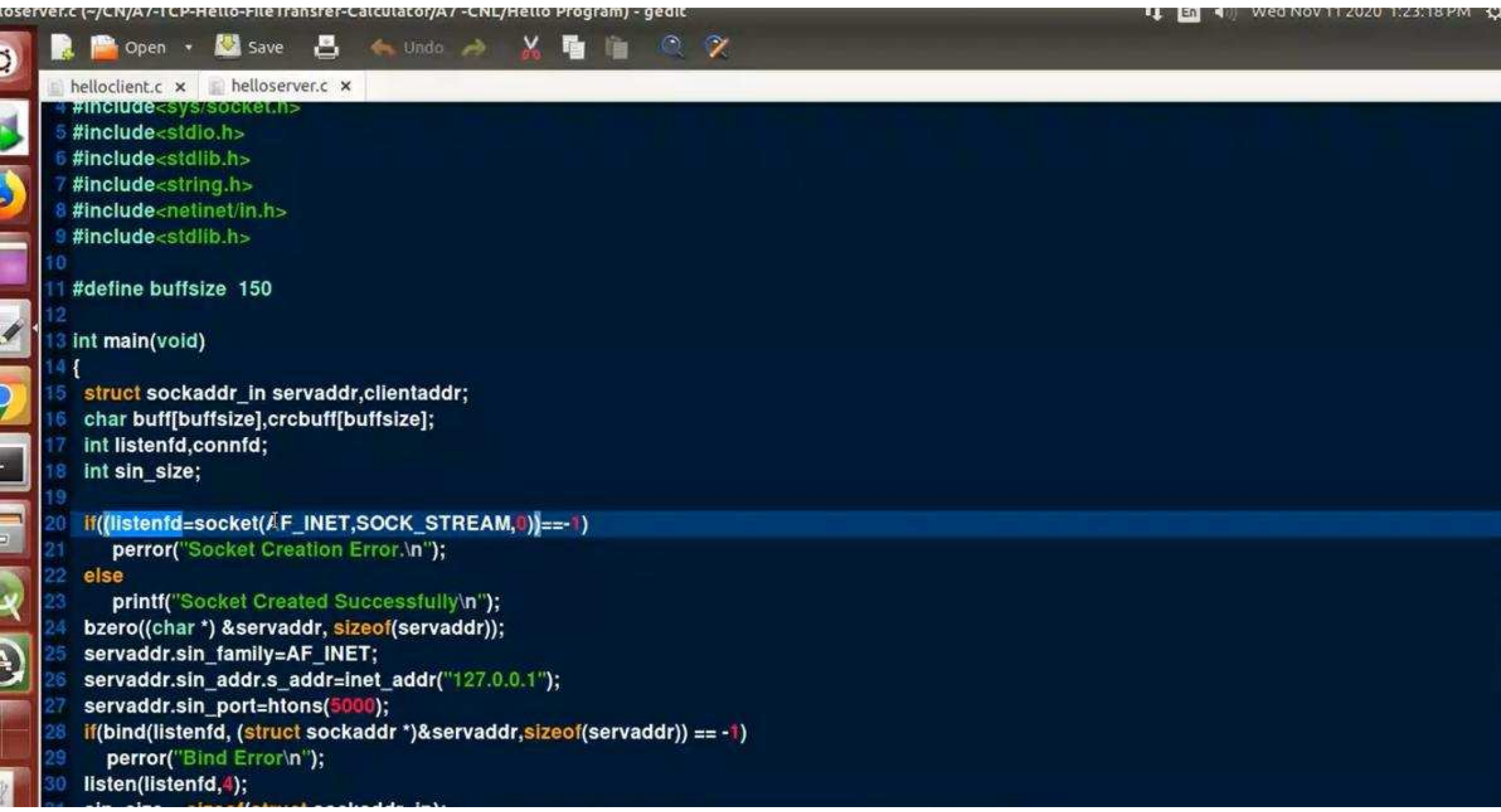
- There is no guarantee that a message sent via a UDP socket will arrive at its destination, and messages can be delivered in a different order.

- A TCP listener is created & starts listening to the specified port.

Conclusion :

Thus we have learned about TCP socket for wired network.

Scanned with CamScanner


helloserver.c (~/CN/A7-TCP-Hello-File Transfer-Calculator/A7 -CNL/Hello Program) - gedit
Open
Save
Undo
helloclient.c
helloserver.c
#include<sys/socket.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<netinet/in.h>
#include<stdlib.h>
#define buffsize 150
int main(void)
{
struct sockaddr_in servaddr,clientaddr;
char buff[buffsize],crcbuff[buffsize];
int listenfd,connfd;
int sin_size;
if((listenfd=socket(AF_INET,SOCK_STREAM,0))==-1)
perror("Socket Creation Error.\n");
else
printf("Socket Created Successfully\n");
bzero((char *) &servaddr, sizeof(servaddr));
servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=inet_addr("127.0.0.1");
servaddr.sin_port=htons(5000);
if(bind(listenfd, (struct sockaddr *)&servaddr,sizeof(servaddr)) == -1)
perror("Bind Error\n");
listen(listenfd,4);

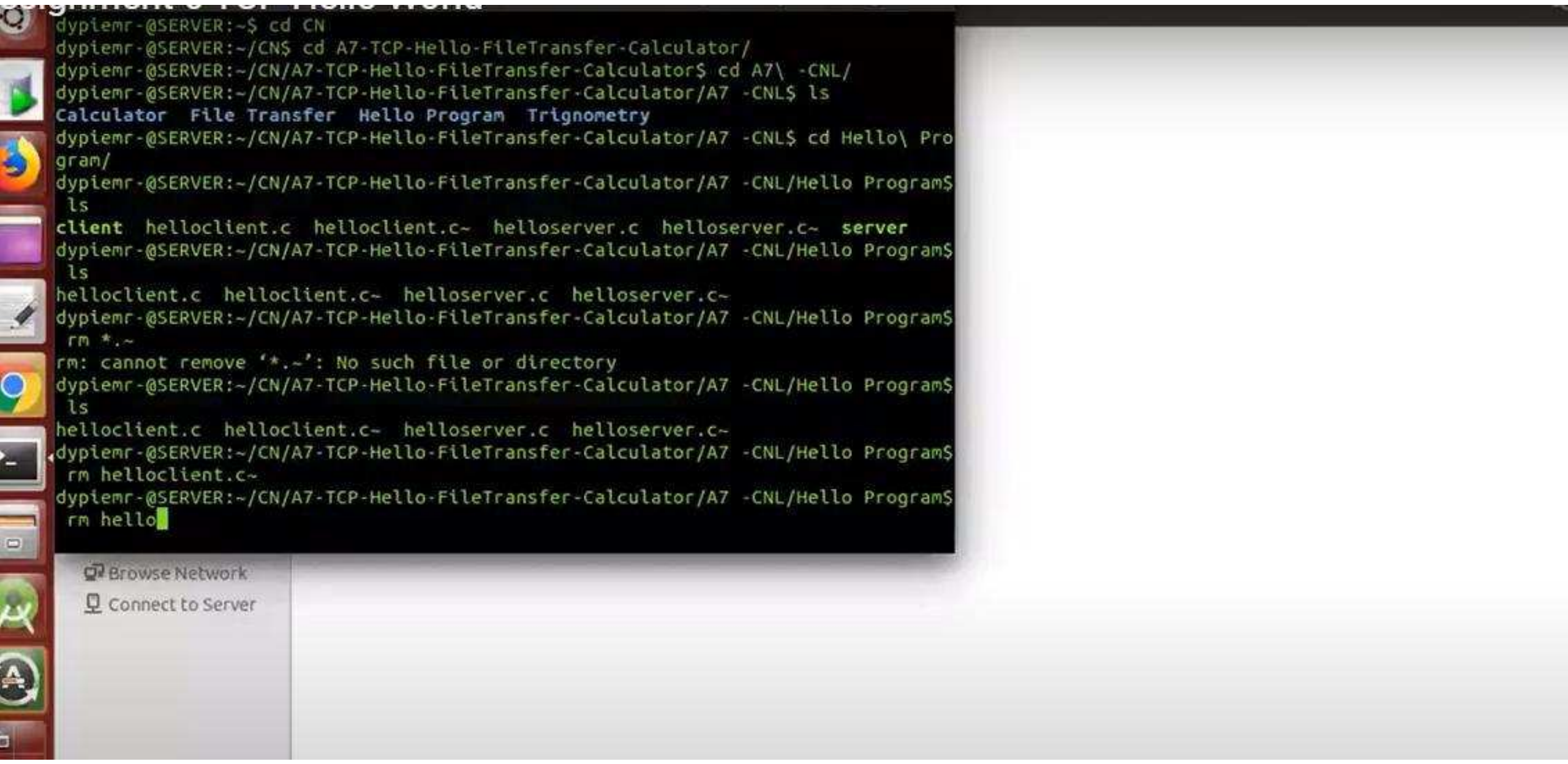
helloclient.c × helloserver.c ×

```c
// Hello Client Side

#include<sys/types.h>
#include<sys/socket.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<netinet/in.h>
 #include<arpa/inet.h>
#include<stdlib.h>

#define buffsize  150

int main(void)
{
 struct sockaddr_in clientaddr;
 char recvline[buffsize];
 int sockfd,n;
 if((sockfd=socket(AF_INET,SOCK_STREAM,0))<0)
     perror("Socket Creation Error.\n");
 else
     printf("Socket Created Successfully\n");
 bzero((char *) &clientaddr, sizeof(clientaddr));
 clientaddr.sin_family=AF_INET;
 clientaddr.sin_addr.s_addr=inet_addr("127.0.0.1");
 clientaddr.sin_port=htons(5000);
 if(connect(sockfd,(struct sockaddr *) &clientaddr, sizeof(clientaddr)) < 0 )          perror("Connect error\n");
```


dypiemr-@SERVER:~$ cd CN
dypiemr-@SERVER:~/CN$ cd A7-TCP-Hello-FileTransfer-Calculator/
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator$ cd A7\ -CNL/
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL$ ls
Calculator File Transfer Hello Program Trignometry
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL$ cd Hello\ Program/
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL/Hello Program$ ls
client helloclient.c helloclient.c~ helloserver.c helloserver.c~ server
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL/Hello Program$ ls
helloclient.c helloclient.c~ helloserver.c helloserver.c~
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL/Hello Program$ rm *.~
rm: cannot remove '*.~': No such file or directory
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL/Hello Program$ ls
helloclient.c helloclient.c~ helloserver.c helloserver.c~
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL/Hello Program$ rm helloclient.c~
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL/Hello Program$ rm hello
Browse Network
Connect to Server

```
 gcc helloserver.c
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL/Hello Program$
 ls
a.out  helloclient.c  helloserver.c
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL/Hello Program$
 ls -l
total 20
-rwxrwxr-x 1 dypiemr- dypiemr- 9146 Nov 11 13:37 a.out
-rw-r--r-- 1 dypiemr- dypiemr-  968 Nov 11 12:58 helloclient.c
-rw-r--r-- 1 dypiemr- dypiemr- 1092 Nov 11 13:29 helloserver.c
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL/Hello Program$
 gcc -o client helloclient.c
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL/Hello Program$
 ls
a.out  client  helloclient.c  helloserver.c
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL/Hello Program$
 ls -l
total 32
-rwxrwxr-x 1 dypiemr- dypiemr- 9146 Nov 11 13:37 a.out
-rwxrwxr-x 1 dypiemr- dypiemr- 9034 Nov 11 13:38 client
-rw-r--r-- 1 dypiemr- dypiemr-  968 Nov 11 12:58 helloclient.c
-rw-r--r-- 1 dypiemr- dypiemr- 1092 Nov 11 13:29 helloserver.c
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL/Hello Program$
```

Browse Network

Connect to Server

```
total 32
-rwxrwxr-x 1 dypiemr- dypiemr- 9146 Nov 11 13:37 a.out
-rwxrwxr-x 1 dypiemr- dypiemr- 9034 Nov 11 13:38 client
-rw-r--r-- 1 dypiemr- dypiemr-  968 Nov 11 12:58 helloclient.c
-rw-r--r-- 1 dypiemr- dypiemr- 1092 Nov 11 13:29 helloserver.c
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL/Hello Program$
 rm a.out
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL/Hello Program$
 ls -l
total 20
-rwxrwxr-x 1 dypiemr- dypiemr- 9034 Nov 11 13:38 client
-rw-r--r-- 1 dypiemr- dypiemr-  968 Nov 11 12:58 helloclient.c
-rw-r--r-- 1 dypiemr- dypiemr- 1092 Nov 11 13:29 helloserver.c
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CN
 gcc -o server helloserver.c
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CN
 ls -l
total 32
-rwxrwxr-x 1 dypiemr- dypiemr- 9034 Nov 11 13:38 client
-rw-r--r-- 1 dypiemr- dypiemr-  968 Nov 11 12:58 helloclient.c
-rw-r--r-- 1 dypiemr- dypiemr- 1092 Nov 11 13:29 helloserver.c
-rwxrwxr-x 1 dypiemr- dypiemr- 9146 Nov 11 13:38 server
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CN
```

Browse Network

Connect to Server

```
dypiemr-@SERVER: ~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL/Hello Program
dypiemr-@SERVER:~$ cd CN
dypiemr-@SERVER:~/CN$ cd A7-TCP-Hello-FileTransfer-Calculator/
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator$ cd A7\ -CNL/
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL$ cd Hello\ Pro
gram/
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL/Hello Program$
 ls -l
total 32
-rwxrwxr-x 1 dypiemr- dypiemr- 9034 Nov 11 13:38 client
-rw-r--r-- 1 dypiemr- dypiemr-  968 Nov 11 12:58 helloclient.c
-rw-r--r-- 1 dypiemr- dypiemr- 1092 Nov 11 13:29 helloserver.c
-rwxrwxr-x 1 dypiemr- dypiemr- 9146 Nov 11 13:38 server
dypiemr-@SERVER:~/CN/A7-TCP-Hello-FileTransfer-Calculator/A7 -CNL/Hello Program$
 ./server
Socket Created Successfully
```

Pratik Pingale 19C00056

Assignment-9

- Title:

Write a program using UDP sockets to enable file transfer (Script, Text, Audio & Video) between two machines. Demonstrate the packets captured traces using wireshark packet analyzer tool for peer to peer mode.

- Requirements:
  - Fedora 20 with pentium IV and above
  - 1 GB RAM
  - 120 GB HDD
  - Monitor
  - Keyboard
  - Mouse
  - Modelio
  - Eclipse
  - CDT
  - Python interpreter
  - Pydev
  - J2SE
  - Wireshark packet analyzer tool.

- Theory:

- TCP guarantees the delivery of packets & preserves their order on destination.

- Sometimes the features are not required & since they do not come without performance costs, it would be better to use a lighter transport protocol.

Scanned with CamScanner

- This kind of service is accomplished by the UDP protocol which conveys datagram packets.

- This kind of service is accomplished by the UDP protocol which conveys datagram packets.

- The format of datagram packet is:

|Msg|length|Host|server Port|

- Java supports datagram communication through the following classes:
  - DatagramPacket
  - DatagramSocket

- The key methods of DatagramPacket class are: byte[] getData() returns the data buffer.

- Packet Types:

  - All unicast packets that are being sent to one of the addresses for that adapter.

  - All multicast packets that are being sent to a multicast address for that adapter.

  - All broadcast packets.

Scanned with CamScanner

- Data Packets:

  - Data packets are often supplied to the packet capture mechanism, by default, as 'fake' ethernet packets.

- Non Data Packets:

  - We have to capture in monitor mode to capture non-data packets.

- Management Packets:

  - Management packets are used by peer WLAN controllers to maintain a WLAN network.

- Low level Control Packets:

  - Control packets are used by peer WLAN controllers to synchronize channel access within contending WLAN hardware.

- 80211 Filter (Modes):

  - Filters packets on the receiving side in several ways.

- Channels (Frequencies):

  - 802.11. adapters will filter packets on the receiving side.

  - 802.11 uses radio frequencies in the range of 2412-2484 MHz.

Scanned with CamScanner

- Monitor mode:

  - In monitor mode the SSID filter mentioned above is disabled and all SSID's from the currently selected channel are captured.

- For Linux System:

  ~~Whether~~ - Newer Linux levels support the mac80211 framework for 802.11 adapter drivers.

  - Drivers listed as supporting cfg80211 & monitor mode should support enough of the mac80211 framework to allow monitor mode to be controlled in a standard fashion.

- Conclusion:

  - Hence we understood how to use wireshark packet analyzer tool.

Scanned with CamScanner

```
int main()
{
  char fileName[100];
  char filebuffer[2000],caufile[maxlen];
  char *vfilep;
  int aufile[700000],vfile[mlen];
  int sd,connfd,len;

for(int i=0;i<=100;i++){
        fileName[i]='\0';
}
  struct sockaddr_in servaddr,cliaddr;



//-----------------------------creation of socket-----------------------------------------
sd = socket(AF_INET, SOCK_DGRAM, 0);

if(sd==-1)
  {
    printf(" socket not created in server\n");
    exit(0);
  }
else
  {
    printf("socket created in  server\n");
  }



bzero(&servaddr, sizeof(servaddr));
```

```
    }
  else
    {
      printf("socket created in  server\n");
    }



  bzero(&servaddr, sizeof(servaddr));


//add the parameters to the socket
  servaddr.sin_family = AF_INET;
  servaddr.sin_addr.s_addr = INADDR_ANY;//for local host
  servaddr.sin_port = htons(8000);//set the port number

  memset(&(servaddr.sin_zero),'\0',8);



//---------------------------bind the socket-----------------------------------------------
  if ( bind(sd, (struct sockaddr *)&servaddr, sizeof(servaddr)) != 0 )
    printf("Not binded\n");
  else
    printf("Binded\n");

  len=sizeof(cliaddr);

  int choice =1;
  while(1)
  {
        char num;
```

server.c

```c
int choice =1;
while(1)
{
        char num;

        recvfrom(sd,&num,sizeof(num),0,(struct sockaddr *)&cliaddr, &len);


        choice = num;

        switch(choice)
        {
                case 1:
                        recvfrom(sd,fileName,1024,0,(struct sockaddr *)&cliaddr, &len);
                        printf("NAME OF TEXT FILE RECEIVED : %s\n",fileName);
                        FILE *fp;
                        printf("Contents in the received text file : \n");
                        recvfrom(sd,filebuffer,1024,0,(struct sockaddr *)&cliaddr, &len);
                        printf("%s\n",filebuffer);
                        int fsize=strlen(filebuffer);
                        fp=fopen(fileName,"w");
                        if(fp)
                        {
                                fwrite(filebuffer, fsize, 1, fp);
                                printf("File received successfully.\n");
                        }
                        else
                        {
                                printf("Cannot create to output file.\n");
                        }
                        memset(fileName, '\0', sizeof(fileName));
                        fclose(fp);
```

```
choice = num;


switch(choice)
{
        case 1:
                recvfrom(sd,fileName,1024,0,(struct sockaddr *)&cliaddr, &len);
                printf("NAME OF TEXT FILE RECEIVED : %s\n",fileName);
                FILE *fp;
                printf("Contents in the received text file : \n");
                recvfrom(sd,filebuffer,1024,0,(struct sockaddr *)&cliaddr, &len);
                printf("%s\n",filebuffer);
                int fsize=strlen(filebuffer);
                fp=fopen(fileName,"w");
                if(fp)
                {
                        fwrite(filebuffer, fsize, 1, fp);
                        printf("File received successfully.\n");
                }
                else
                {
                        printf("Cannot create to output file.\n");
                }
                memset(fileName, '\0', sizeof(fileName));
                fclose(fp);
                break;
        case 2:
                recvfrom(sd,fileName,1024,0,(struct sockaddr *)&cliaddr, &len);
                printf("NAME OF AUDIO FILE RECEIVED : %s\n",fileName);
                FILE *afp;
                int numbytes;
```

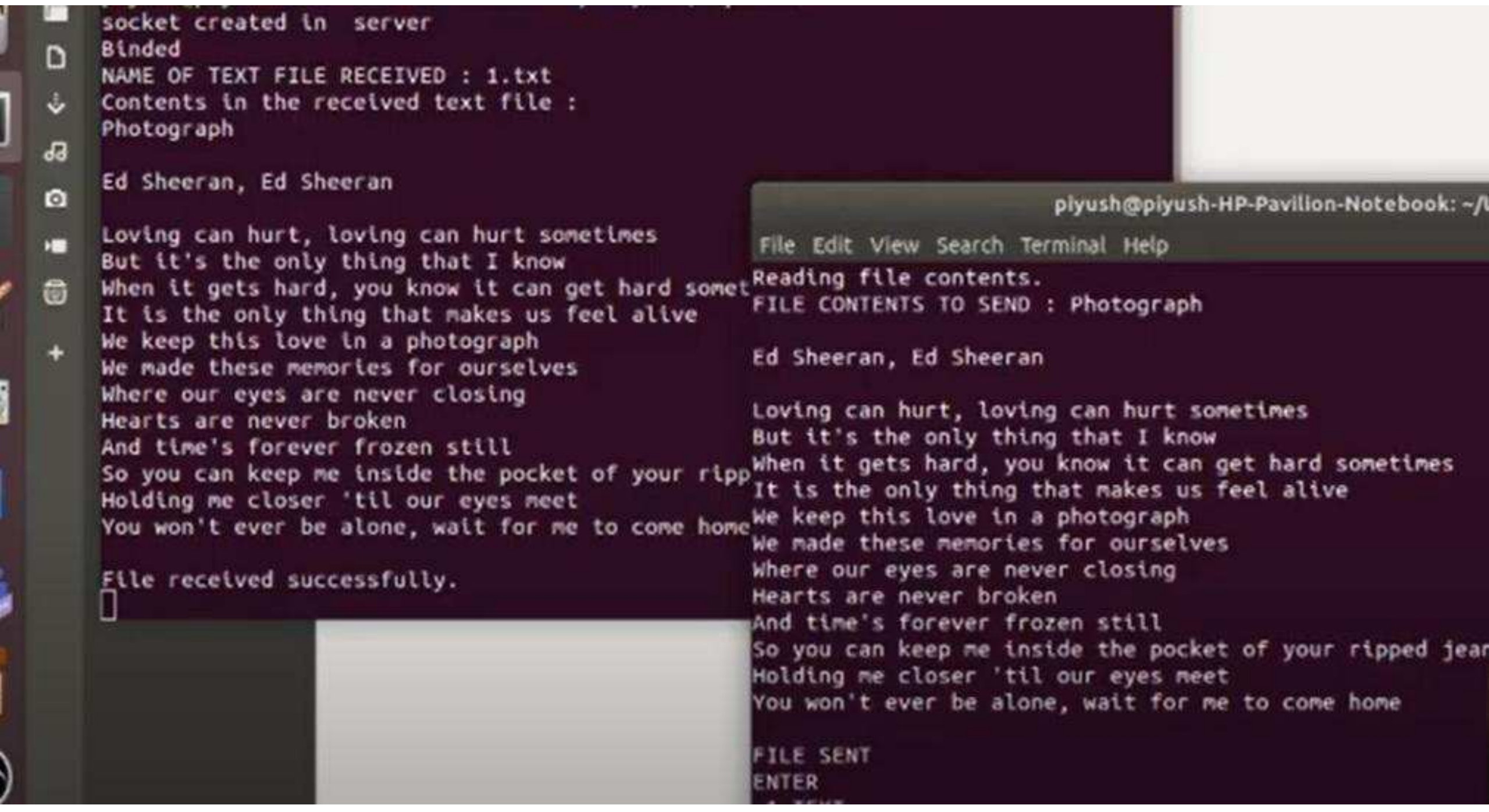
```
*server.c
        {
                printf("Cannot open output file.\n");
        }
        memset(fileName, '\0', sizeof(fileName));
        fclose(afp);
        break;

case 3:
        recvfrom(sd,fileName,1024,0,(struct sockaddr *)&cliaddr, &len);
        printf("VIDEO FILE NAME RECEIVED : %s\n",fileName);
        FILE *vfp;
        vfp=fopen(fileName,"w");
        size_t  vfsize;
        vfsize=recvfrom(sd,vfile,100000,0,(struct sockaddr *)&cliaddr, &len);

        if(vfp)
        {
                fwrite(vfile, vfsize, 1, vfp);
                printf("File received successfully.\n");
        }
        else
        {
                printf("Cannot open output file.\n");
        }
        fclose(vfp);
        break;

case 4:
        close(sd);
        break;

    }
}
```


socket created in server
Binded
NAME OF TEXT FILE RECEIVED : 1.txt
Contents in the received text file :
Photograph
Ed Sheeran, Ed Sheeran
Loving can hurt, loving can hurt sometimes
But it's the only thing that I know
When it gets hard, you know it can get hard somet
It is the only thing that makes us feel alive
We keep this love in a photograph
We made these memories for ourselves
Where our eyes are never closing
Hearts are never broken
And time's forever frozen still
So you can keep me inside the pocket of your ripp
Holding me closer 'til our eyes meet
You won't ever be alone, wait for me to come home
File received successfully.
piyush@piyush-HP-Pavilion-Notebook: ~/
File Edit View Search Terminal Help
Reading file contents.
FILE CONTENTS TO SEND : Photograph
Ed Sheeran, Ed Sheeran
Loving can hurt, loving can hurt sometimes
But it's the only thing that I know
When it gets hard, you know it can get hard sometimes
It is the only thing that makes us feel alive
We keep this love in a photograph
We made these memories for ourselves
Where our eyes are never closing
Hearts are never broken
And time's forever frozen still
So you can keep me inside the pocket of your ripped jean
Holding me closer 'til our eyes meet
You won't ever be alone, wait for me to come home
FILE SENT
ENTER

Pratik Pingale 19C0056

Assignment - 10

- Title:

Write a program for DNS lookup. Given an IP address input, it should return URL and vice-versa.

- Outcomes:

Retrieve DNS from Given IP and vice-versa.

- Theory:

- The DNS Hierarchy:

- The DNS Namespace is based on a 'tree' structure, with a small number of generic Top level domains and a large no. of country based domains.

- Each TLD supports a group of 'second level domains' and so on.

- Resource Records:

- Resource records are 5 tuples:

Domain_name TTL TYPE CLASS VALUE

- Domain_name:

- The name of the domain to which this RR applies.

- TTL:

- The time to live of this RR.

Scanned with CamScanner

- Type:

  - There are several types of RR such as,
    - SOA
    - IP address
    - NS

- Class:

  - Always set to 'IN', for Internet

- Value:

  - The actual value of this particular RR.

- DSS servers & Resolvers:

  - A nameserver provides domain-name-to-IP address mappings 'looking up' for one or more zones, which are sub-trees of the domain space.

  - Exactly which server is responsible for a particular zone is specified in SOA RRs.

  - It also has a email address of the site administrator, a unique serial number.

  - The DNS system forms a distributed database of domain information.

Scanned with CamScanner

-A resolver is a library function which queries the name server when called from a user program.

- Conclusion:

-Hence we have studied the concept of DNS

Scanned with CamScanner

```
C:\Users\dan>
C:\Users\dan>nslookup
Default Server:  DD-WRT
Address:  192.168.1.1

> server 8.8.8.8
Default Server:  google-public-dns-a.google.com
Address:  8.8.8.8

> sheridan.edu
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

Non-authoritative answer:
Name:    sheridan.edu
Address:  137.89.64.16

> ns1.sheridan.edu
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

Non-authoritative answer:
Name:    ns1.sheridan.edu
Addresses:  72.175.255.235
          137.89.70.15

> set type=mx
> sheridan.edu
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

Non-authoritative answer:
sheridan.edu    MX preference = 10, mail exchanger = 528187713.mail.outlook.com
sheridan.edu    MX preference = 0, mail exchanger = mail.sheridan.edu
sheridan.edu    MX preference = 1, mail exchanger = mail2.sheridan.edu
sheridan.edu    MX preference = 5, mail exchanger = mailex.sheridan.edu
```

```
C:\Users\dan>
C:\Users\dan>nslookup
Default Server:  DD-WRT
Address:  192.168.1.1

> server 8.8.8.8
Default Server:  google-public-dns-a.google.com
Address:  8.8.8.8

> sheridan.edu
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

Non-authoritative answer:
Name:    sheridan.edu
Address:  137.89.64.16

> ns1.sheridan.edu
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

Non-authoritative answer:
Name:    ns1.sheridan.edu
Addresses:  72.175.255.235
          137.89.70.15

> set type=mx
> sheridan.edu
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

Non-authoritative answer:
sheridan.edu    MX preference = 10, mail exchanger = 528187713.mail.outlook.com
sheridan.edu    MX preference = 0, mail exchanger = mail.sheridan.edu
sheridan.edu    MX preference = 1, mail exchanger = mail2.sheridan.edu
sheridan.edu    MX preference = 5, mail exchanger = mailex.sheridan.edu
> _
```

```
sheridan.edu    MX preference = 10, mail exchanger = 528187713.mail.outlook.com
sheridan.edu    MX preference = 0, mail exchanger = mail.sheridan.edu
sheridan.edu    MX preference = 1, mail exchanger = mail2.sheridan.edu
sheridan.edu    MX preference = 5, mail exchanger = mailex.sheridan.edu
> set type=ptr
> 137.89.64.16
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

Non-authoritative answer:
16.64.89.137.in-addr.arpa       name = admiral.gc.cc.wy.us
16.64.89.137.in-addr.arpa       name = www.bigwestartsfestival.com
16.64.89.137.in-addr.arpa       name = diablo.gc.cc.wy.us
> set type a
Unrecognized command: set type a
> set type=a
> sheridan.edu
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

Non-authoritative answer:
Name:    sheridan.edu
Address:  137.89.64.16

> ^C
C:\Users\dan>nslookup yahoo.com
Server:  DD-WRT
Address:  192.168.1.1

Non-authoritative answer:
Name:    yahoo.com
Addresses:  72.30.2.43
          98.137.149.56
          209.191.122.70
          67.195.160.76
          69.147.125.65
```

Assignment - 11

- Title:

Installing & configure DHCP server & write a program to Install the software on remote machine.

- Theory:

- DHCP stands for Dynamic Host Configuration Protocol.

- DHCP is a standardized network protocol used on IP networks

- DHCP Server can be any server that is used to distribute IP addresses automatically to the clients in the network.

- DHCP is a network service that enables host computers to be automatically assigned settings from a server.

The most common settings provided by a DHCP server to DHCP clients include:

- IP address & netmask

- IP address of the default gateway to use

- IP address of the DNS servers to use.

- However a DHCP can also configuration properties such as:
  - Host name
  - Domain name
  - Time server
  - Print server

Scanned with CamScanner

- Advantage of using DHCP is that changes to the network, need only to be changed at DHCP server.

- It is also easier to integrate new computers into the network.

- Manual Allocation:

  - Entails using DHCP to identify the unique hardware address of each network card connected to the network & the continue supplying constant configuration.

- Dynamic Allocation:

  - In this, DHCP server will assign IP address from a pool of addresses for a period of time or less, that is configured using the server until the client informs the server that it doesn't need it.

- Automic Allocation:

  - Using this, DHCP automatically assign an IP address permanently to a device, selecting it from a pool of ~~device~~ available addresses.

- Install DHCP Server in Ubuntu:

  - Enter the following command:

    sudo apt-get install isc-dhcp-server

Scanned with CamScanner

- Configuration :

  - DHCP server configuration is no that difficult.

  - We have to assign on what interface should the DHCP server serve DHCP requests.

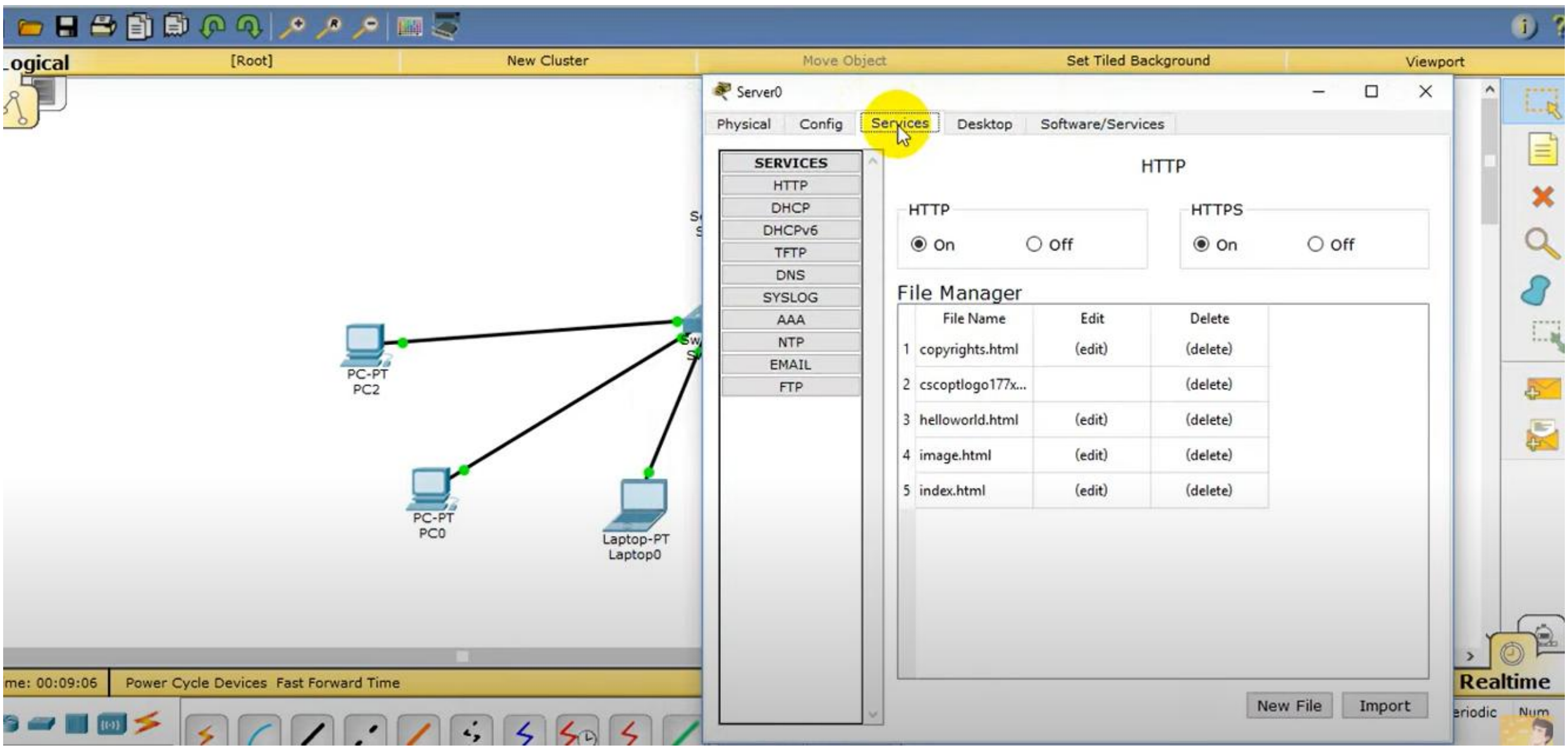
  - Assign the network interface.

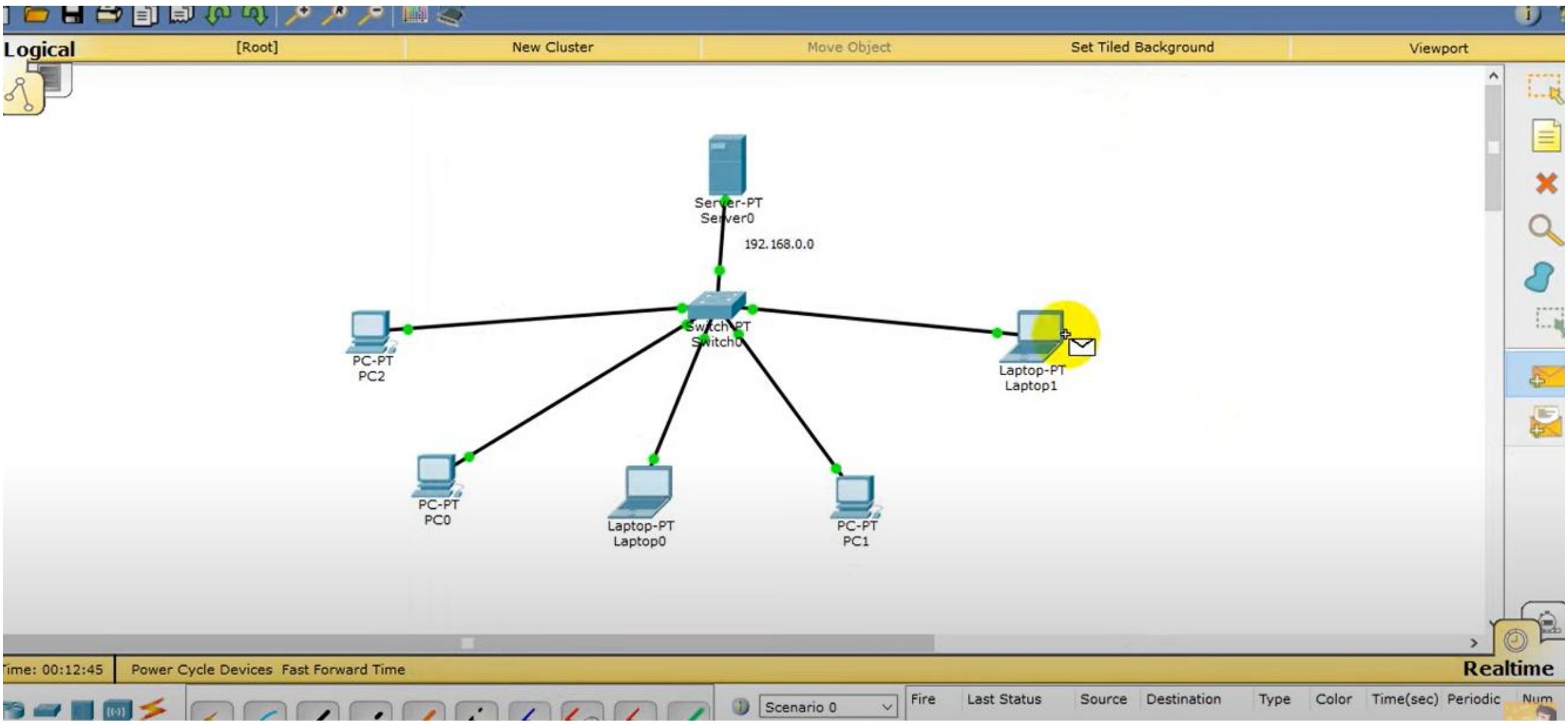
  - Now, go to the client configuration network settings & change IP settings to automatic (DHCP)

- Conclusion :

  Hence we have learned about DHCP server

Scanned with CamScanner


[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Server0
Physical
Config
Services
Desktop
Software/Services
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
HTTP
HTTP
On
Off
HTTPS
On
Off
File Manager
File Name
Edit
Delete
1 copyrights.html (edit) (delete)
2 cscoptlogo177x... (delete)
3 helloworld.html (edit) (delete)
4 image.html (edit) (delete)
5 index.html (edit) (delete)
New File
Import
PC-PT
PC2
PC-PT
PC0
Laptop-PT
Laptop0
Power Cycle Devices
Fast Forward Time
Realtime


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Server-PT
Server0
192.168.0.0
Switch-PT
Switch0
PC-PT
PC2
Laptop-PT
Laptop1
PC-PT
PC0
Laptop-PT
Laptop0
PC-PT
PC1
Time: 00:12:45
Power Cycle Devices
Fast Forward Time
Realtime
Scenario 0
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num

Pratik Pingale 19C0056

Assignment - 12

- Title :

Study and analyze the performance of HTTP, HTTPS & FTP protocol.

- Outcomes :

Configuration of HTTP, HTTPS and FTP protocol in CISCO packet tracer.

- Theory:

1. HTTP :

- Hypertext Transfer Protocol is an application-level protocol for distributed, collaborative, hypermedia extension systems.

- This is foundation for data communication for WWW since 1990.

- It is a generic and stateless protocol which is used for other purposes as well

- It is a TCP/IP based communication protocol

- Client :

- HTTP Client sends a request to the server in the form of a request method.

Scanned with CamScanner

• Server:

- HTTP server responds with a status line, including the message's protocol version.

2. HTTPS :

- When properly configured, an HTTPS connection guarantees :

i) confidentality

ii) authenticity

iii) integrity

- A plain HTTP connection can be easily monitored, modified & impersonated.

- HTTPS established an encrypted link between the browser & the web server using SSL & TLS.

3. FTP :

- FTP is standard internet protocol provided by TCP/IP

- Mainly used for transmitting the web page files from their creator to computer.

- Also used for downloading the files to computers.

Scanned with CamScanner

• Objectives :

- Provides sharing of files
- Used to encourage use of remote computers
- Transfers the data more reliably and efficiently.

• There are two types of connections in FTP:

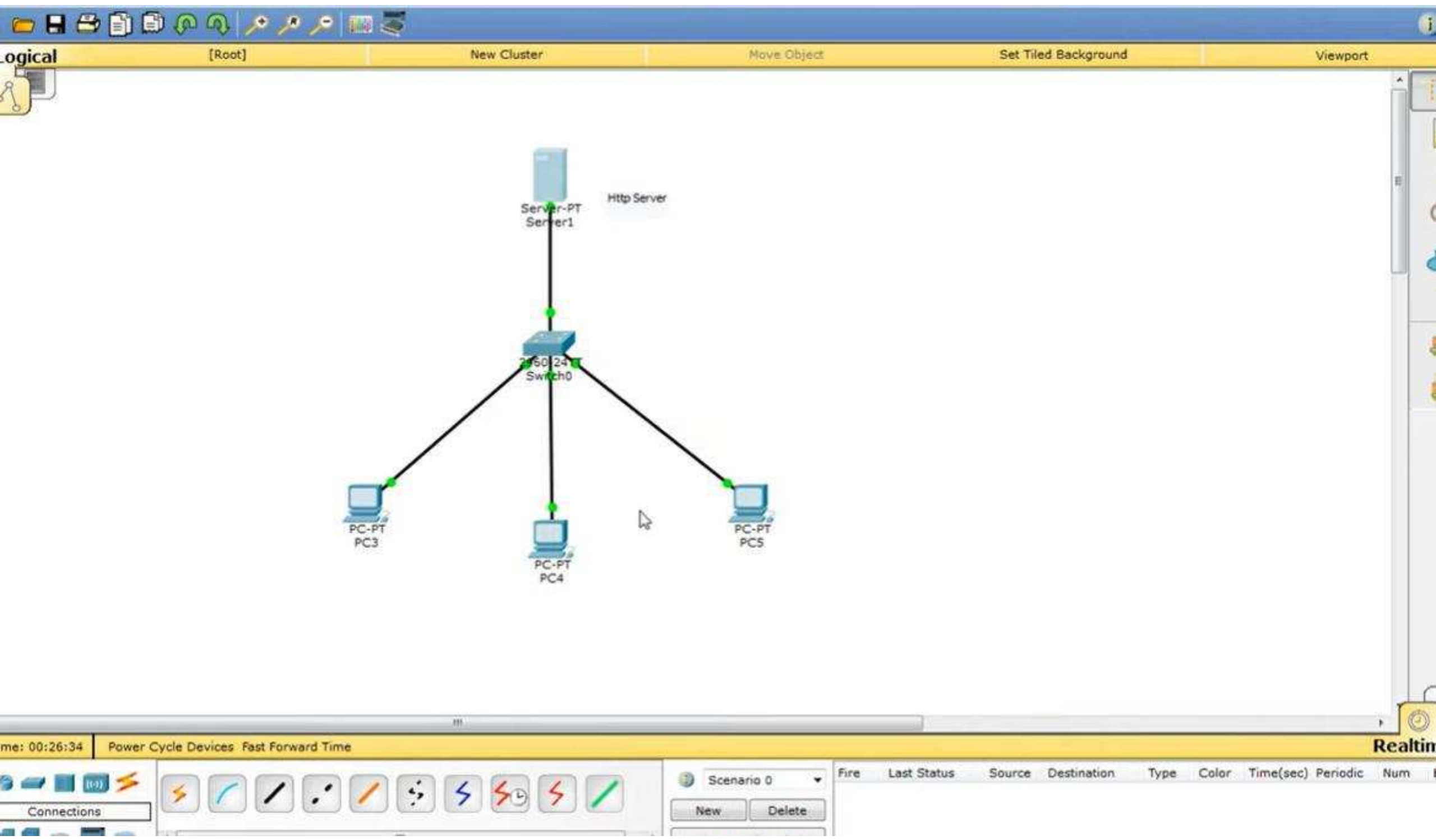
1) Control Connection:
   - It uses very simple rules for communication.

2) Data Connection:
   - Uses complex rules as data types may vary.

• Conclusion:

- Configure HTTP, HTTPS & FTP protocol in CISCO packet tracer.

Scanned with CamScanner


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Server-PT
Server1
Http Server
2960-24TT
Switch0
PC-PT
PC3
PC-PT
PC4
PC-PT
PC5
Power Cycle Devices
Fast Forward Time
Connections
Scenario 0
New
Delete
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num

Pratik Pingale 19C0056

Assignment - 13

Title :

To study the SSL protocol by capturing the packets using wireshark tool while visiting any SSL secured website.

Outcomes :

Retrieve SSL secured protocol by capturing the packets using wireshark.

Theory :

- SSL :

- Secure socket layer is an encryption based internet security protocol

- It was first developed by Netscape in 1995 for purpose of ensuring privacy, authentication & data integrity.

- How does SSL/TLS Work?

- In order to provide a high degree of privacy.

- SSL encrypts data that is transmitted across the web.

- SSL initiates an authentication process called a handshake between both devices.

- Also digitally signs data in order to provide data integrity.

- There have been several iterations of SSL, each more secure than the last.

Scanned with CamScanner

- Why is SSL / TLS important?

  - Originally data on the web was transmitted in plaintext that anyone could read if they intercepted the message.

  - SSL was created to correct this problem & protect user privacy.

  - By encrypting any data that use between a user & a web server.

  - SSL also stops certain kinds of cyber attacks.

  - Also prevents data tempering.

- Are SSL & TLS the same thing?

  - Since they are so closely related, the two are often interchanged & confused.

  - Some people still use SSL refer to TLS others use "SSL / TLS encryption" because SSL still has so much name recognisitio

- Conclusion:

  - Thus we have studied SSL and wireshark.

Scanned with CamScanner

```
Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : localdomain
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Network Connection
   Physical Address. . . . . . . . . : 00-0C-29-AB-5D-16
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::a1ab:c058:f003:7695%11(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.170.183(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Tuesday, May 28, 2013 8:49:12 AM
   Lease Expires . . . . . . . . . . : Tuesday, May 28, 2013 10:04:12 AM
   Default Gateway . . . . . . . . . : 192.168.170.2
   DHCP Server . . . . . . . . . . . : 192.168.170.254
   DHCPv6 IAID . . . . . . . . . . . : 234884137
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-19-13-57-B7-00-0C-29-AB-5D-1

   DNS Servers . . . . . . . . . . . : 192.168.170.2
   Primary WINS Server . . . . . . . : 192.168.170.2
   NetBIOS over Tcpip. . . . . . . . : Enabled

Tunnel adapter isatap.{C8C22CA0-0D10-4CB0-BA24-F1722F1CF48D}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Microsoft ISATAP Adapter
   Physical Address. . . . . . . . . : 00-00-00-00-00-00-00-E0
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes

Tunnel adapter Local Area Connection* 9:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Teredo Tunneling Pseudo-Interface
   Physical Address. . . . . . . . . : 00-00-00-00-00-00-00-E0
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   IPv6 Address. . . . . . . . . . . : 2001:0:9d38:6ab8:2494:6a2:3f57:5548(Pref
erred)
   Link-local IPv6 Address . . . . . : fe80::2494:6a2:3f57:5548%13(Preferred)
   Default Gateway . . . . . . . . . : ::
   NetBIOS over Tcpip. . . . . . . . : Disabled

Tunnel adapter isatap.localdomain:
```

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
2266	6.02079500	74.125.225.109	192.168.170.183	TCP	60	https > 49379 [FIN, PSH, ACK] Seq=6098 Ack=1499 Win=6423
2267	6.02080800	192.168.170.183	173.194.46.94	TCP	54	49373 > https [ACK] Seq=3619 Ack=19274 Win=63643 Len=0
2268	6.02085800	192.168.170.183	173.194.46.47	TCP	54	49403 > https [ACK] Seq=1090 Ack=4238 Win=64240 Len=0
2269	6.02089300	192.168.170.183	74.125.225.109	TCP	54	49379 > https [ACK] Seq=1499 Ack=6099 Win=63630 Len=0
2270	6.02283200	74.125.225.33	192.168.170.183	TCP	60	https > 49367 [FIN, PSH, ACK] Seq=1 Ack=2 Win=64239 Len=
2271	6.02284400	192.168.170.183	74.125.225.33	TCP	54	49367 > https [ACK] Seq=2 Ack=2 Win=63735 Len=0
2272	6.02294500	173.194.46.47	192.168.170.183	TCP	60	https > 49369 [FIN, PSH, ACK] Seq=1 Ack=2 Win=64239 Len=
2273	6.02295600	192.168.170.183	173.194.46.47	TCP	54	49369 > https [ACK] Seq=2 Ack=2 Win=63646 Len=0
2274	6.02693200	74.125.192.95	192.168.170.183	TCP	60	https > 49372 [FIN, PSH, ACK] Seq=2745 Ack=916 Win=64239
2275	6.02694500	192.168.170.183	74.125.192.95	TCP	54	49372 > https [ACK] Seq=916 Ack=2746 Win=63453 Len=0
2276	6.03528500	74.125.142.95	192.168.170.183	TCP	60	https > 49366 [FIN, PSH, ACK] Seq=1 Ack=2 Win=64239 Len=
2277	6.03531700	192.168.170.183	74.125.142.95	TCP	54	49366 > https [ACK] Seq=2 Ack=2 Win=63174 Len=0
2278	6.03871300	74.125.225.179	192.168.170.183	TCP	60	https > 49368 [FIN, PSH, ACK] Seq=94729 Ack=1872 Win=642
2279	6.03874000	192.168.170.183	74.125.225.179	TCP	54	49368 > https [ACK] Seq=1872 Ack=94730 Win=64240 Len=0
2280	11.5399340	192.168.170.1	192.168.170.255	DB-LSP-	165	Dropbox LAN sync Discovery Protocol
2281	41.5559770	192.168.170.1	192.168.170.255	DB-LSP-	165	Dropbox LAN sync Discovery Protocol

- Frame 1: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface 0
- Ethernet II, Src: Vmware_ab:5d:16 (00:0c:29:ab:5d:16), Dst: Vmware_e5:86:c6 (00:50:56:e5:86:c6)
- Internet Protocol Version 4, Src: 192.168.170.183 (192.168.170.183), Dst: 192.168.170.2 (192.168.170.2)
- User Datagram Protocol, Src Port: 64091 (64091), Dst Port: domain (53)
- Domain Name System (query)

```
0000  00 50 56 e5 86 c6 00 0c  29 ab 5d 16 08 00 45 00   .PV..... ).]...E.
0010  00 3f 35 7e 00 00 80 11  00 00 c0 a8 aa b7 c0 a8   .?5~.... ........
0020  aa 02 fa 5b 00 35 00 2b  d6 47 60 e4 01 00 00 01   ...[.5.+ .G`.....
0030  00 00 00 00 00 00 06 63  68 72 6f 6d 65 06 67 6f   .......c hrome.go
0040  6f 67 6c 65 03 63 6f 6d  00 00 01 00 01            ogle.com .....
```

Intel(R) PRO/1000 MT Network Connection: \D | Packets: 2281 Displayed: 2281 Marked: 0 | Profile: Default

Pratik Pingale 19CO056

Assignment - 14

- Title :

To study the IPsec protocol by capturing the packets using wireshark tool.

- Outcomes :

Retrieve IPsec protocol by capturing the packets using wireshark tool.

- Theory :

• IPsec :

- The IP security is an internet engineering task force (IETF) standard suite protocols between 2 communication points across the IP network.

- It also defines the encrypted, decrypted & authenticated packets.

• What Ports does IPsec operate on?:

- UDP port 500 should be opened as should IP protocols 50 & 51.

- UDP port 500 should be opened to allow for ISAKMP to be forwarded through the firewall.

Scanned with CamScanner

- ISAKMP:

  - ISAKMP stands for Internet Security Association key management Protocol.

  - Key components of an IPsec VPN.

- What are ESP and AH?

  - ESP stands for Encapsulating Security Protocol.

  - AH stands for Authentication Header.

- Encapsulating Security Protocol:

  - ESP gives protection to upper layer new protocols.

  - Unless a data packet is being tunneled, ESP protects only the IP data payload & not IP header.

- Authentication Header:

  - AH is a new protocol & part of the IPsec protocol suite

  - If security associations have bein established, AH can be optionally configured to defend against replay attacks.

- How Do They All work Together?

  - When properly configured, an IPsec VPN provides multiple layers of security that ensure the security mode and integrity of that is being transmitted through the encrypted tunnel.

Scanned with CamScanner

- Conclusion:

  - We have studied IPsec protocol.

Scanned with CamScanner