Medi-Caps University, Indore

Department of Computer Science Engineering

Computer Networks

CS3C012

SUBMITTED BY-

Dhananjay Porwal

EN19CS301110

SUBMITTED TO-

Mr. Bhupendra Panchal

Experiment 1

1. Study and design of various cables and connectors.

Cable Media

There are a number of different cabling types that a network engineer/administrator sees over the course of their career. New individuals coming in to the field have to be familiar with a number of different cables and connectors to be prepared for their day-to-day activities. The following cabling media types will be most commonly seen in LAN environments.

Unshielded Twisted Pair (UTP)

The most common type of cable is Unshielded Twisted Pair (UTP); as the name suggests, this type of cable includes an unshielded media that includes twisted pairs. Specifically, UTP includes four different pairs of copper cables that are each twisted together; the twisted rate depends on the specific category of cable. Figure 1 shows an example of UTP cabling:

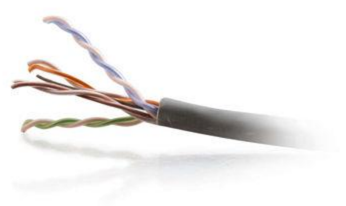


Figure 1 Unshielded Twisted Pair Cabling

Most networks in the last 15 years or so have included UTP cabling that is at least rated as category 5; this cabling supports Ethernet data rates up to 100 Mbps. On modern networks that have been deployed recently, it is common to install at least a category 6 cable with support for up to 1000 Mbps; Table 1 shows a short list of the available UTP categories and their supported rates.

Table 1: UTP Cabling Categories

Category	Frequency Bandwidth	Typical Technologies Supported
5	100 MHz	100 Mbps Ethernet (100-Base-TX)
5e	100 MHz	100 Mbps Ethernet (100-Base-TX) and 1000 Mbps Ethernet (1000-Base-T)
6	250 MHz	1000 Mbps Ethernet (1000-Base-T)
6a	500 MHz	1000 Mbps Ethernet (1000-Base-T) and 10 Gbps Ethernet (10GBase-T)

Shielded Twisted Pair (STP)

Another type of twisted pair cabling includes a shield that is used to both contain and protect the different twisted pairs from interference. This type of cabling is not commonly seen in modern LAN networks deployments at the time of this writing; however some newer very high speed Ethernet technologies may require a cable that has a shield and thus will be shown here.

The term STP can include a number of different cable types which all include a shielding mechanism. Some cable types include a shield only between the different twisted pairs within the cable and others include various shielding types both around the pairs and the whole cable; the specifics will not be covered in this article. Figure 2 shows an example of an STP cable that has a shield between the pairs and the whole cable:

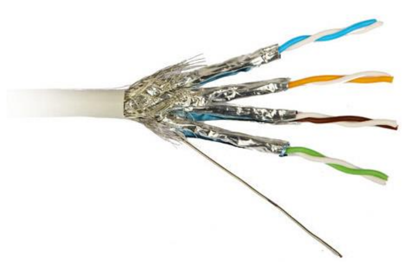


Figure 2 Shielded Twisted Pair Cabling

Multi-Mode Fiber (MMF)

A very common type of fiber connection is Multi-Mode fiber (MMF); this type of cable uses light to transmit signals between devices and is thus not susceptible to electrical interference. MMF cables use a larger internal core diameter (typically, 50 µm or 62.5 µm) and can utilize lower cost LEDs for transmission; this is both an advantage and a disadvantage. While the larger core diameter offers a cable that supports multiple modes and a cable that is easier to work with (light coming into the cable is allowed to come in at multiple angles), it is also limited by the same factors in terms of total useable cable length. MMF cables are typically only used for connections that are less than 2 kilometers in length; this also makes it a very common cable in LAN deployments. Figure 3 below shows an example of a MMF cable:

Figure 3 Multi-Mode Fiber (MMF) Cable

Single Mode Fiber (SMF)

Like Multi-Mode Fiber (MMF), Single Mode Fibers (SMF) transmits signals via light and is not subject to electrical interference. The difference between SMF and MMF is in their physical characteristics; a MMF cable has a large core diameter and is able to accept a number of different modes that come into the cable from multiple angles, SMF has a much smaller core diameter (typically 8-10 µm) and accepts signals coming in from a specific angle and on a specific mode. The specifics of how the light propagate within the cable are really outside the scope of required knowledge of an entry level network engineer/administrator; what does need to be known is that MMF is typically used for shorter cable runs (up to 2 km typically) and SMF can be used for cable runs of very long distances (typically up to ~40 miles without repeaters depending on wavelength). Figure 4 below shows an example of a SMF fiber:

Figure 4 Single-Mode Fiber (SMF) Cable

Connectors

With all the different types of cabling come a number of different cable connectors. This section takes a look at the most common cabling connectors.

Registered Jack 45 (RJ45)

The cable connector that is found on almost all UTP and STP cables is a Registered Jack 45 which is mostly commonly referred to as RJ45. This type of connector resembles the older RJ11 connectors that most people are familiar with from wired telephones. Figure 5 below shows an example of a RJ45 connector:

Figure 5 Registered Jack-45 (RJ45) Connector

Straight Tip (ST)

The Straight Tip (ST) connector is often seen on the end of a multi-mode cable; it has been commonly seen along with the SC connector for the last 20 years but is being slowly replaced by multi-fiber connectors (LC and MTP). Figure 6 below shows an example of a ST connector:

Figure 6 Straight Tip (ST) Connector

Subscriber Connector (SC)

The Subscriber Connector (SC) can be seen commonly on MMF or SMF; as with SC connectors, the ST connector is slowly being replaced by multi-fiber connectors. Figure 7 below shows an example of an SC connector:

Figure 7 Subscriber Connector (SC)

Lucent Connector (LC)

The Lucent Connector (LC) was developed for high-density deployments where multiple fibers would be terminated within a confined space. Unlike the SC and ST connectors, the LC connector is always duplex connecting a pair of fibers at a time. Figure 8 below shows an example of a LC connector:

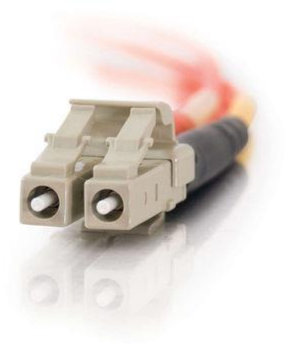


Figure 8 Lucent Connector (LC)

Multi-fiber Push On (MPO)

The Multi-fiber Push On (MPO) connector is another duplex connector that offers an easy options for connection. As the name suggests, it was designed to be able to be connected multiple times without the creation of any potential connector issues. It is often also referred to as Multi-fiber Termination Push-on (MTP); the MTP connector is a brand name (US Conec). Figure 9 below shows an example of an MPO connector:

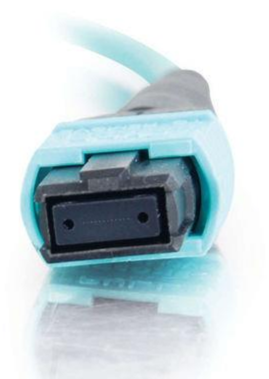


Figure 9 Multi-fiber Push On (MPO) Connector

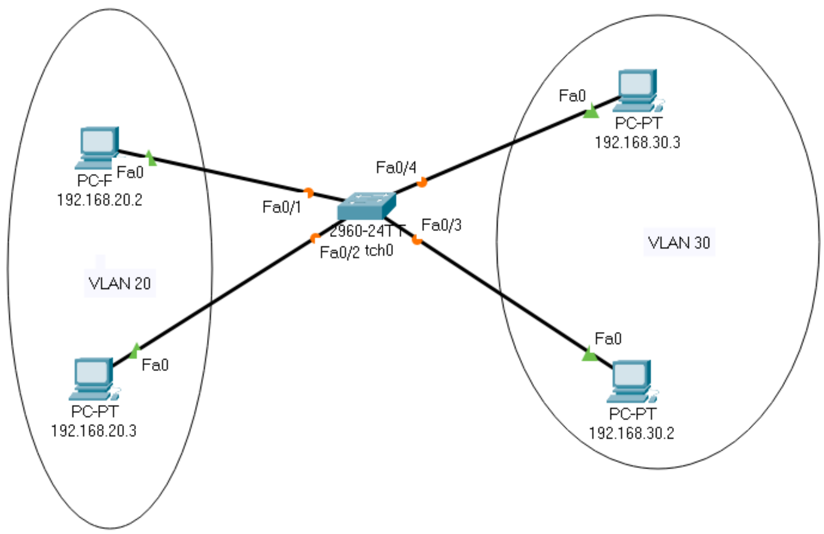
Coaxial Cable Connectors

The most common type of connector used with coaxial cables is the Bayone-Neill-Concelman (BNC) connector. Different types of adapters are available for BNC connectors, including a T-connector, barrel connector, and terminator. Connectors on the cable are the weakest points in any network. To help avoid problems with your network, always use the BNC connectors that crimp, rather screw, onto the cable.

Experiment 2

2.Design of VLAN using layer 2 switch.

```
config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#vlan 20
Switch(config-vlan)#exit
Switch(config)#vlan 30
Switch(config-vlan)#exit
Switch(config)#exit
Switch#


Switch#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#interface f0/1
Switch(config-if)#switchport access vlan 20
Switch(config-if)#exit
Switch(config)#interface fa0/2
Switch(config-if)#switchport access vlan 30
Switch(config-if)#exit
Switch(config)#interface fa0/3
Switch(config-if)#switchport access vlan 30
Switch(config-if)#interface fa0/4
Switch(config-if)#switchport access vlan 30
Switch(config-if)#interface fa0/2
Switch(config-if)#switchport access vlan 20
Switch(config-if)#exit
Switch(config)#show vlan
```



Experiment 3

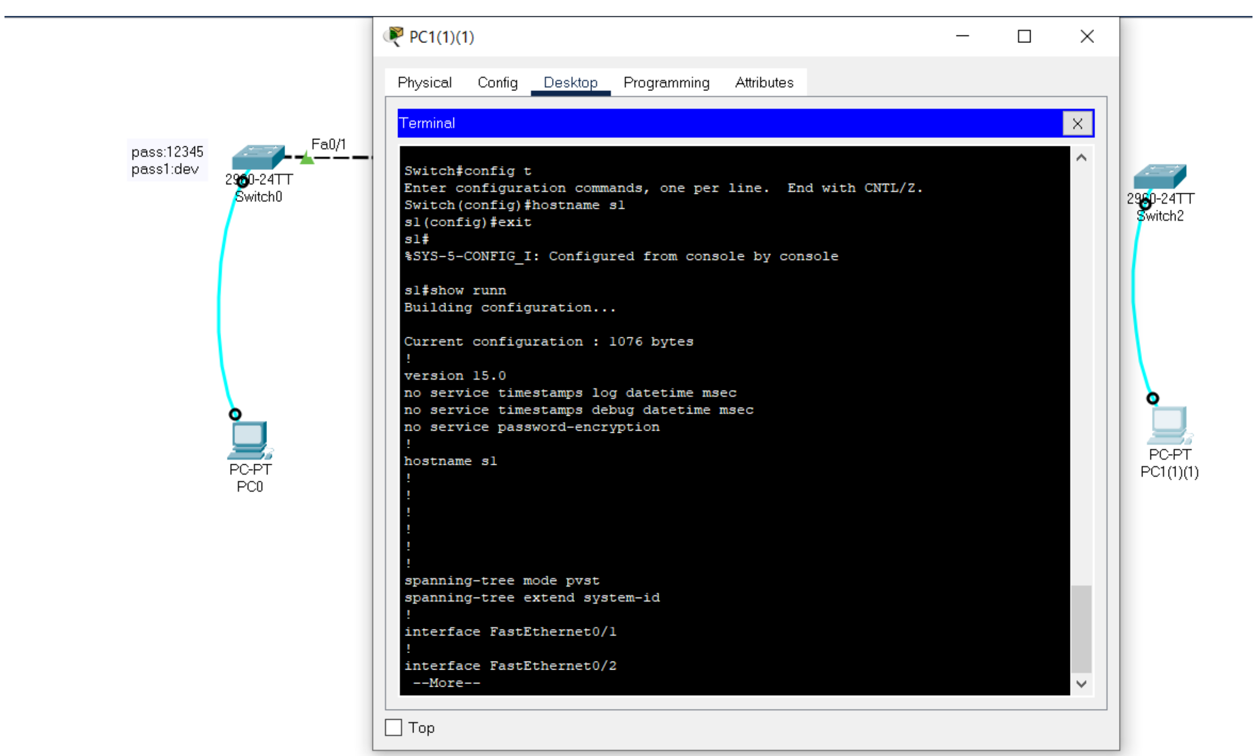
3.Basic Configuration of Non-manageable Layer 2 devices.

Switch>enable
Switch#config t
Switch(config)#hostname s1
s1(config)#exit

%SYS-5-CONFIG_I: Configured from console by console

s1#show runn
Building configuration...

Current configuration : 1076 bytes
!
version 15.0
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname s1
!



Experiment 4

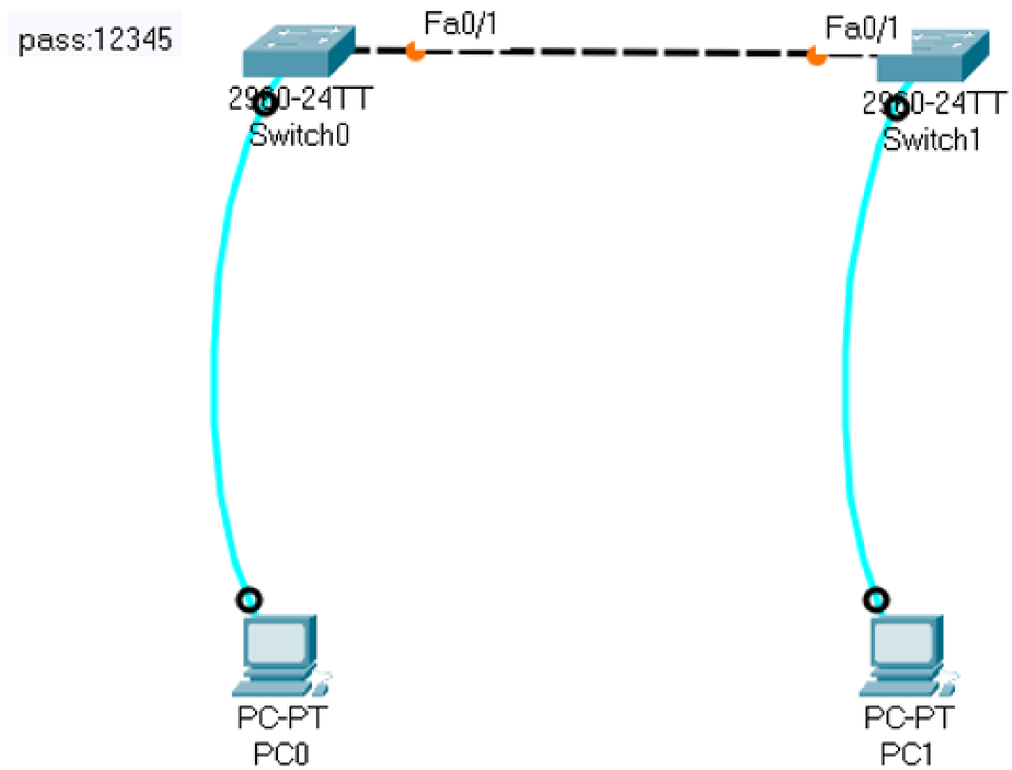
4.Configuration of catalyst switch name password protection various cable used to connect.

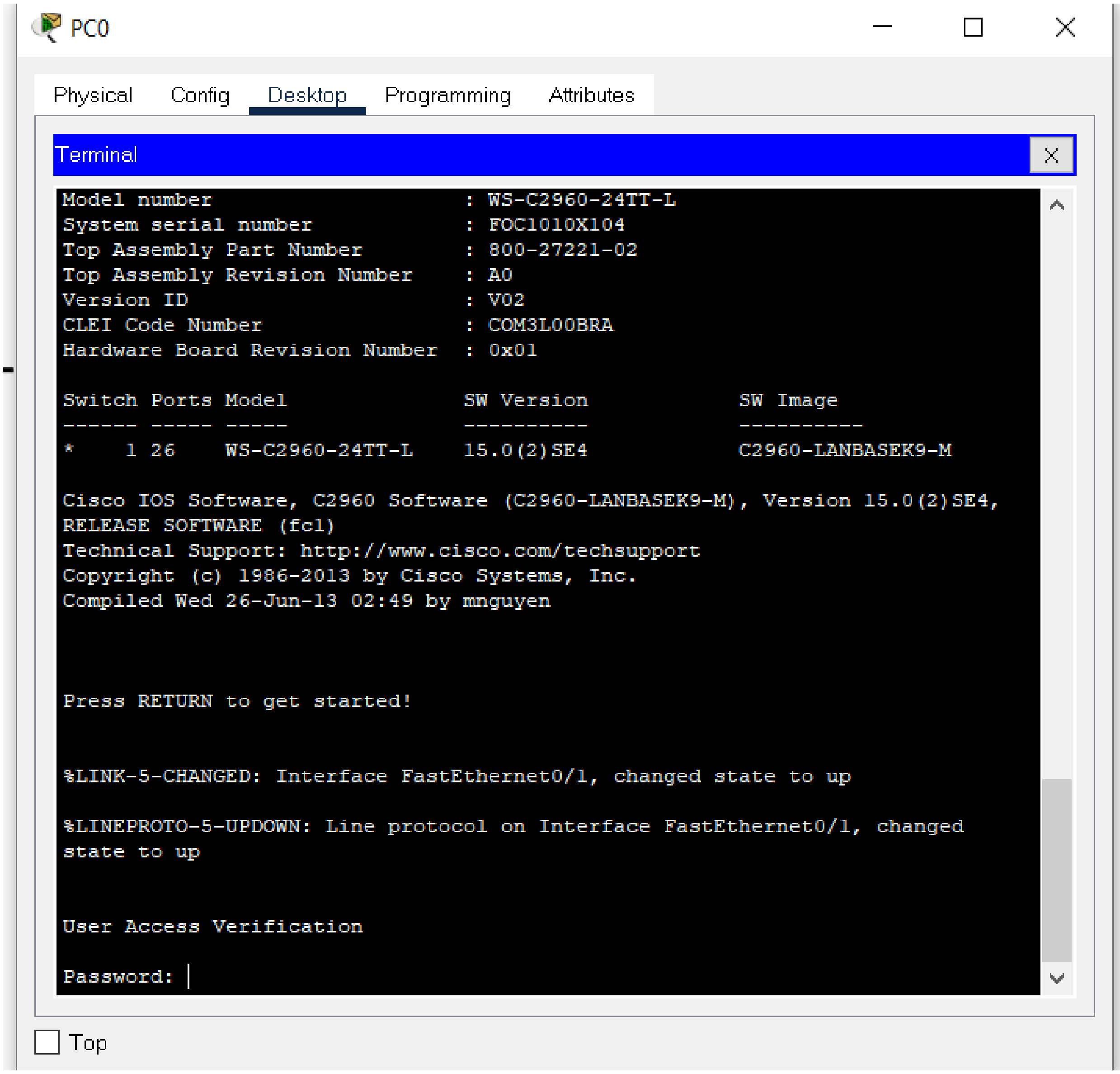
Switch>enable
Switch#config t
Switch(config)#hostname s1
s1(config)#exit
s1#config t
Enter configuration commands, one per line. End with CNTL/Z.
s1(config)#line console 0
s1(config-line)#password 12345
s1(config-line)#login
s1(config-line)#exit
s1(config)#exit
s1#
s1>enable
Password:
s1#config t
Enter configuration commands, one per line. End with CNTL/Z.

```
s1(config)#enable password dev
s1(config)#exit
s1#
%SYS-5-CONFIG_I: Configured from console by console

s1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
s1(config)#enable secret dev123
s1(config)#exit
s1#
%SYS-5-CONFIG_I: Configured from console by console

s1#show runn
Building configuration...

Current configuration : 1181 bytes
!
version 15.0
no service timestamps log datetime msec
no service timestamps debug datetime msec
service password-encryption
!
hostname s1
!
enable secret 5 $1$mERr$Td3IUoGuNutrsTlxfT78F1
enable password 7 08254958
```

same code in CLI of 2nd switch.

We can also use service password-encryption in configuration mode to encrypt password.


pass:12345
Fa0/1
Fa0/1
2960-24TT
Switch0
2960-24TT
Switch1
PC-PT
PC0
PC-PT
PC1


PC0
Physical
Config
Desktop
Programming
Attributes
Terminal
Model number : WS-C2960-24TT-L
System serial number : FOC1010X104
Top Assembly Part Number : 800-27221-02
Top Assembly Revision Number : A0
Version ID : V02
CLEI Code Number : COM3L00BRA
Hardware Board Revision Number : 0x01
Switch Ports Model SW Version SW Image
------ ----- ----- ---------- ----------
* 1 26 WS-C2960-24TT-L 15.0(2)SE4 C2960-LANBASEK9-M
Cisco IOS Software, C2960 Software (C2960-LANBASEK9-M), Version 15.0(2)SE4, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2013 by Cisco Systems, Inc.
Compiled Wed 26-Jun-13 02:49 by mnguyen
Press RETURN to get started!
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
User Access Verification
Password:
Top

Experiment 5

5.Configuration of DHCP using Layer manageable devices.

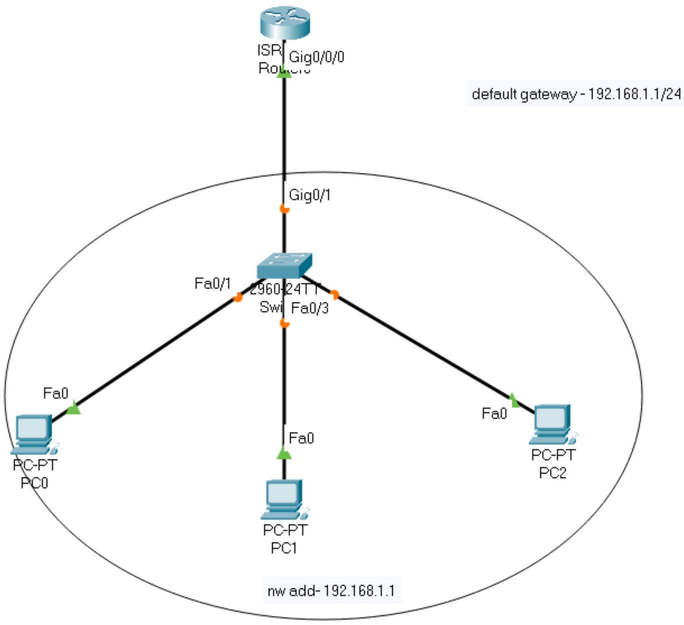
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface g0/0/0
Router(config-if)#ip address 192.168.1.1 255.255.255.0

Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0, changed state to up

Router(config-if)#exit
Router(config)#ip dhcp pool n1
Router(dhcp-config)#network 192.168.1.0 255.255.255.0
Router(dhcp-config)#default-router 192.168.1.1
Router(dhcp-config)#dns-server 192.168.1.1
Router(dhcp-config)#exit
Router(config)#exit
Router#



Experiment 6

Dhananjay Porwal
EN19CS301110

6.Design and configuration of inter VLAN routing using Layer 3 switch.

give ip AND DEFAULT GATEWAYS

@new vlan setup

```
Switch>ENABLE
Switch#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#interface fa0/1
Switch(config-if)#switchport access vlan 10
% Access VLAN does not exist. Creating vlan 10
Switch(config-if)#exit
Switch(config)#vlan 10
Switch(config-vlan)#exit
Switch(config)#vlan 20
Switch(config-vlan)#exit
Switch(config)#vlan 30
Switch(config-vlan)#exit
Switch(config)#interface fa0/1
Switch(config-if)#switchport access vlan 10
Switch(config-if)#switchport mode access
Switch(config-if)#exit
Switch(config)#interface fa0/2
Switch(config-if)#switchport access vlan 20
Switch(config-if)#switchport mode access
Switch(config-if)#exit
Switch(config)#interface fa0/3
Switch(config-if)#switchport access vlan 30
Switch(config-if)#switchport mode access
Switch(config-if)#exit
Switch(config)#exit
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#interface g0/1
Switch(config-if)#no shutdown
Switch(config-if)#switchoport mode trunk

Switch(config-if)#switchport mode trunk
Switch(config-if)#exit
```

```
Switch(config)#exit
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#show runn
Building configuration...
```

@router configuration

```
Router>enable
Router#config t
Router(config)#interface fa0/0
Router(config-if)#no sh

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#interface g0/0.10
%Invalid interface type and number
Router(config)#
Router(config)#interface g0/0/0.10
                     ^
% Invalid input detected at '^' marker.

Router(config)#interface fa0/0.10
Router(config-subif)#
%LINK-5-CHANGED: Interface FastEthernet0/0.10, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.10, changed state to up

Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip address 192.168.10.1 255.255.255.0
Router(config-subif)#exit
Router(config)#interface fa0/0.20
Router(config-subif)#
%LINK-5-CHANGED: Interface FastEthernet0/0.20, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.20, changed state to up

Router(config-subif)#encapsulation dot1Q 20
```

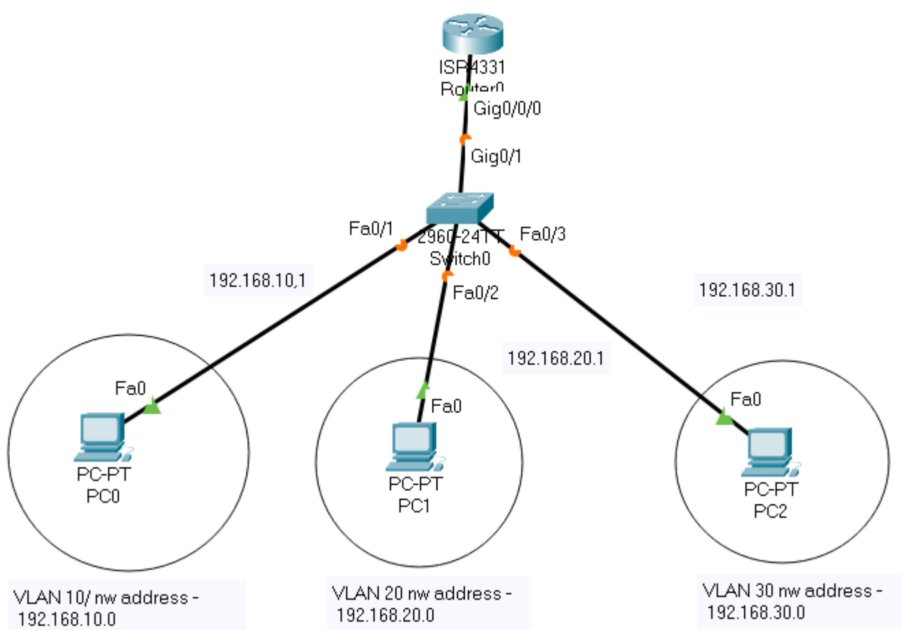
Router(config-subif)#ip address 192.168.20.1 255.255.255.0
Router(config-subif)#exit
Router(config)#interface fa0/0.30
Router(config-subif)#
%LINK-5-CHANGED: Interface FastEthernet0/0.30, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.30, changed state to up

Router(config-subif)#ip address 192.168.30.1 255.255.255.0

% Configuring IP routing on a LAN subinterface is only allowed if that
subinterface is already configured as part of an IEEE 802.10, IEEE 802.1Q,
or ISL vLAN.

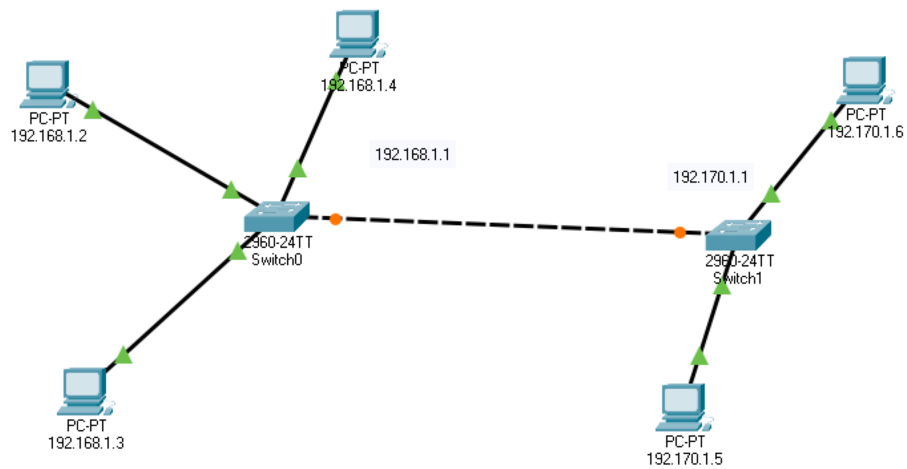
Router(config-subif)#encapsulation dot1Q 30
Router(config-subif)#ip address 192.168.30.1 255.255.255.0
Router(config-subif)#exit

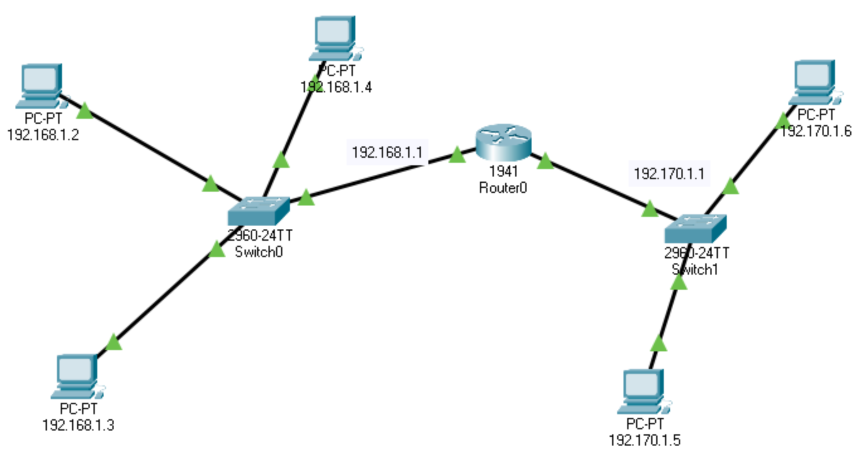


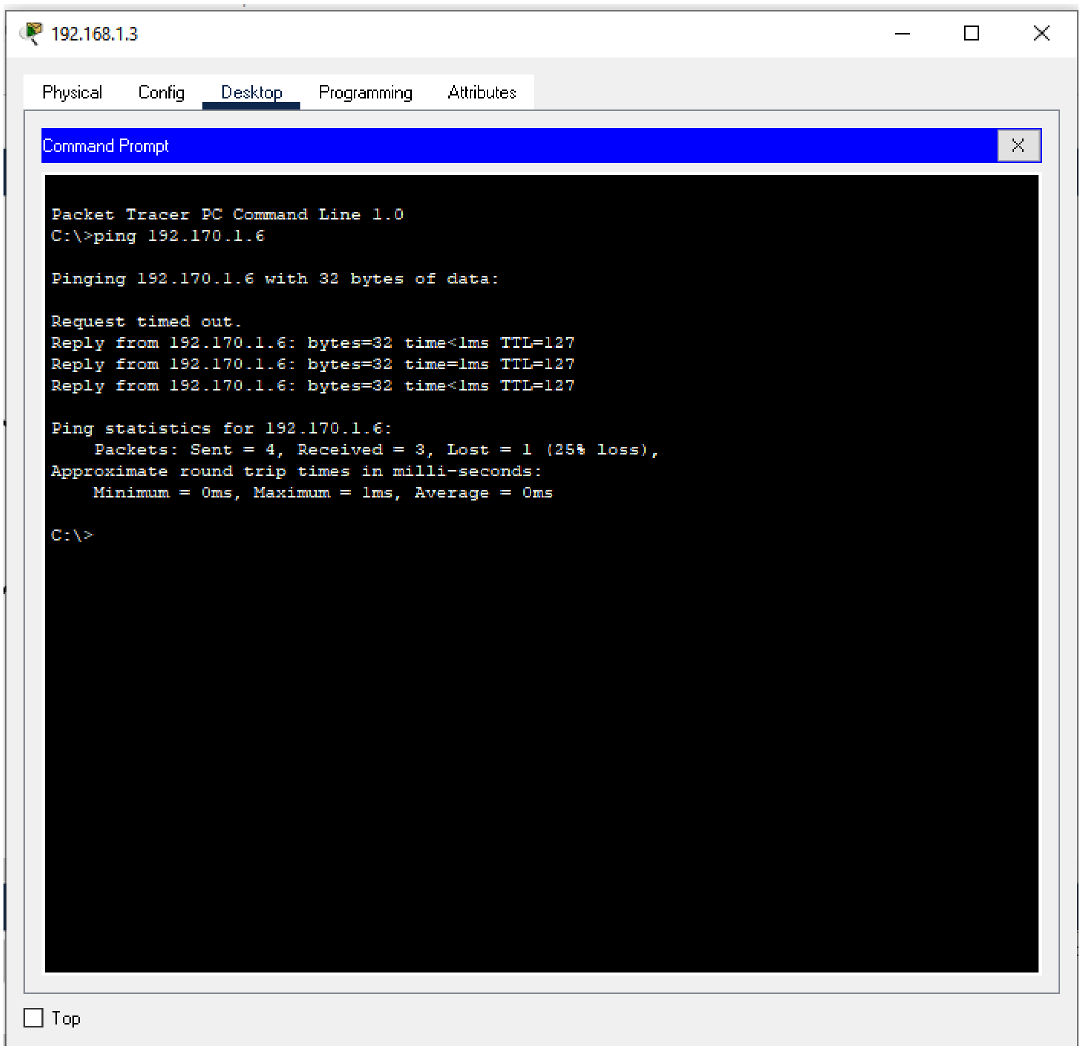
Experiment 7

7.Basic configuration of router and its interfaces.

```
Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R1
R1(config)#interface g0/0/0


R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#no sh

R1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0, changed state to
up

R1(config-if)#exit
R1(config)#interface g0/0/1


R1(config-if)#ip address 192.170.1.1 255.255.255.0
R1(config-if)#no sh

R1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0, changed state to
up

R1(config-if)#exit
R1#
```

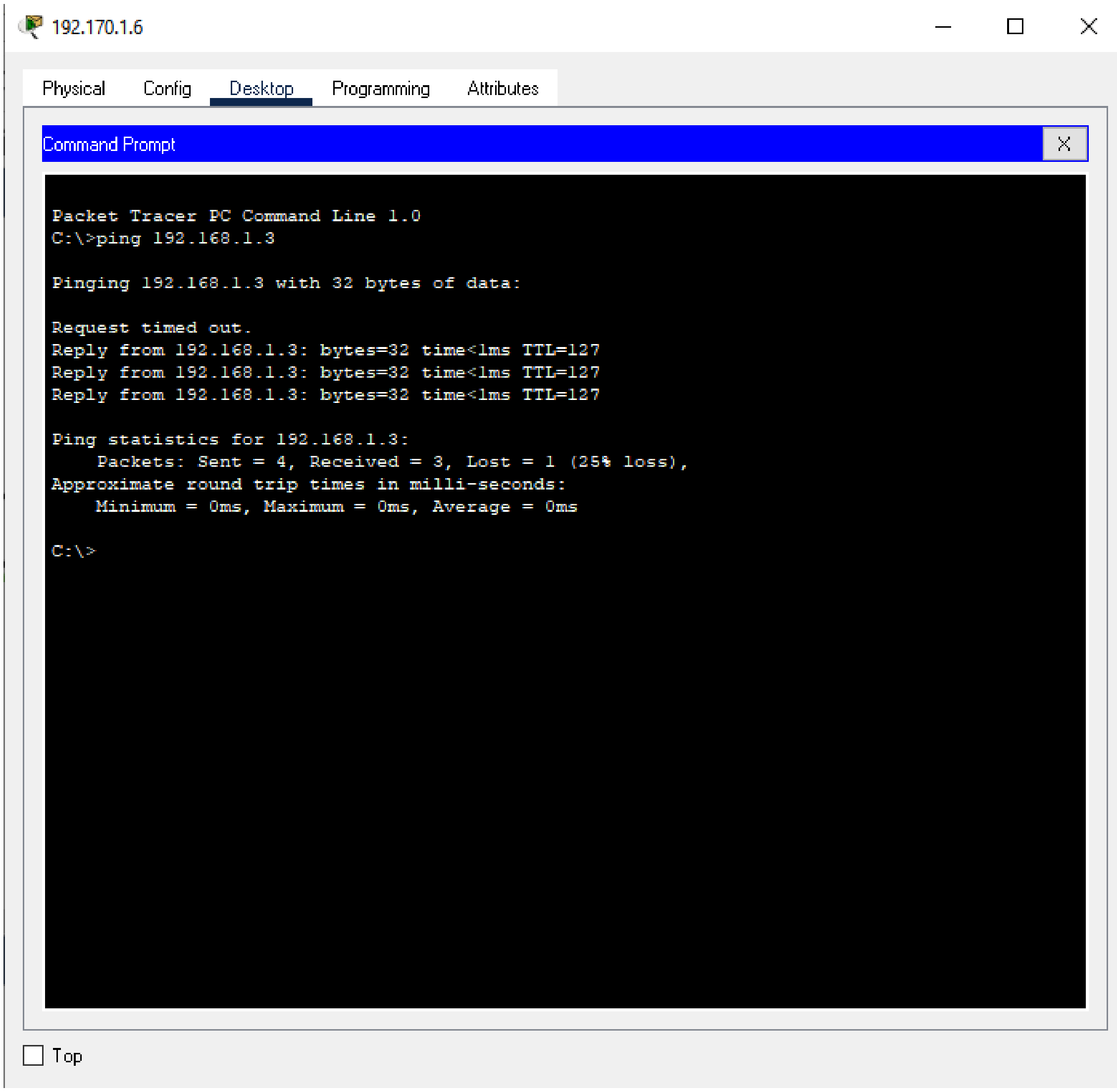

PC-PT
192.168.1.2
PC-PT
192.168.1.4
192.168.1.1
192.170.1.1
PC-PT
192.170.1.6
2960-24TT
Switch0
2960-24TT
Switch1
PC-PT
192.168.1.3
PC-PT
192.170.1.5


PC-PT
192.168.1.2
PC-PT
192.168.1.4
192.168.1.1
1941
Router0
192.170.1.1
PC-PT
192.170.1.6
2960-24TT
Switch0
2960-24TT
Switch1
PC-PT
192.168.1.3
PC-PT
192.170.1.5



:

Pinging PC of 1st network



from PC of 2nd network:

Experiment 8

8.Configuration of DHCP VLAN and inter VLAN routing using router.

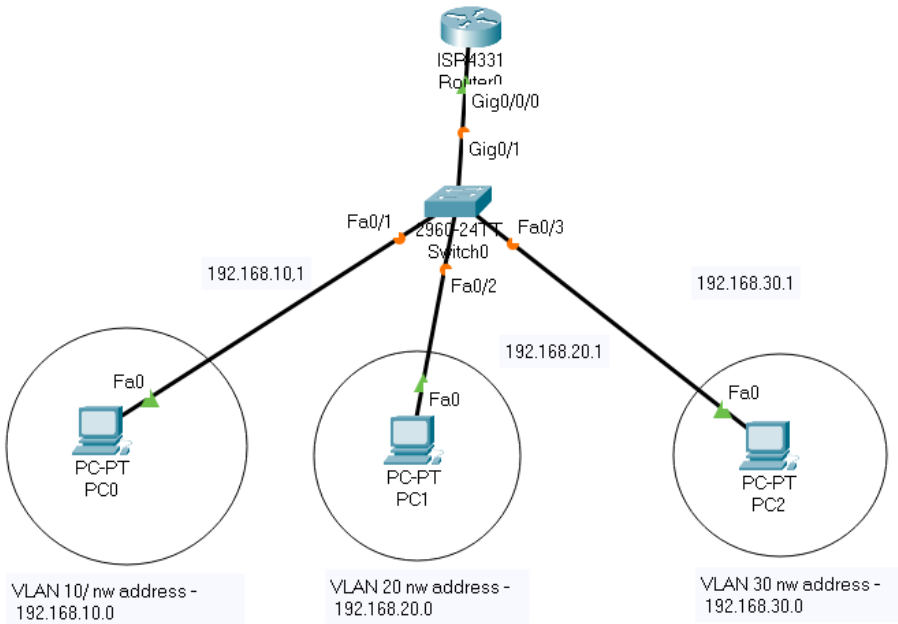
in switch cli

```
Switch>enable
Switch#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#vlan 10
Switch(config-vlan)#exit
Switch(config)#vlan 20
Switch(config-vlan)#exit
Switch(config)#vlan 30
Switch(config-vlan)#exit
Switch(config)#exit
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#show vlan
Switch#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#interface f0/1
Switch(config-if)#switchport access vlan 10
Switch(config-if)#switchport mode access
Switch(config-if)#exit
Switch(config)#interface f0/2
Switch(config-if)#switchport access vlan 20
Switch(config-if)#switchport mode access
Switch(config-if)#exit
Switch(config)#interface f0/3
Switch(config-if)#
Switch(config-if)#switchport access vlan 30
Switch(config-if)#switchport mode access
Switch(config-if)#exit
Switch(config)#interface g0/1
Switch(config-if)#no shutdown
Switch(config-if)#switchport mode trunk
Switch(config-if)#exit
Switch(config)#exit
Switch#
```

in router cli

```
Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface g0/0/0
Router(config-if)#no sh

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0, changed state to
up

Router(config-if)#exit
Router(config)#interface g0/0/0.10
Router(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0.10, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0.10, changed state to
up

Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip address 192.168.10.1 255.255.255.0
Router(config-subif)#exit
Router(config)#interface g0/0/0.20
Router(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0.20, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0.20, changed state to
up

Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip address 192.168.20.1 255.255.255.0
Router(config-subif)#exit
Router(config)#interface g0/0/0.30
Router(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0.30, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0.30, changed state to
up
```

```
Router(config-subif)#encapsulation dot1Q 30
Router(config-subif)#ip address 192.168.30.1 255.255.255.0
Router(config-subif)#exit
Router(config)#


Router(config)#ip dhcp pool n10
Router(dhcp-config)#network 192.168.10.0 255.255.255.0
Router(dhcp-config)#default-router 192.168.10.1
Router(dhcp-config)#dns-server 192.168.10.1
Router(dhcp-config)#exit
Router(config)#ip dhcp pool n20
Router(dhcp-config)#network 192.168.20.0 255.255.255.0
Router(dhcp-config)#dns-server 192.168.20.1

Router(dhcp-config)#default-router 192.168.20.1
Router(dhcp-config)#exit
Router(config)#ip dhcp pool n30
Router(dhcp-config)#network 192.168.30.0 255.255.255.0
Router(dhcp-config)#default-router 192.168.30.1
Router(dhcp-config)#dns-server 192.168.30.1
Router(dhcp-config)#exit
Router(config)#exit
Router#
```


ISR4331
Router0
Gig0/0/0
Gig0/1
2960-24TT
Switch0
Fa0/1
Fa0/2
Fa0/3
192.168.10.1
192.168.20.1
192.168.30.1
Fa0
Fa0
Fa0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
VLAN 10/ nw address -
192.168.10.0
VLAN 20 nw address -
192.168.20.0
VLAN 30 nw address -
192.168.30.0

Use of PING ,TRACERT,NSLOOKUP and various commands used in networking.

ipconfig

The “ipconfig” displays the current information about your network such as your your IP and MAC address, and the IP address of your router. It can also display information about your DHCP and DNS servers. Let’s see the basic output of “ipconfig”:

```
C:\>ipconfig

FastEthernet0 Connection:(default port)

   Connection-specific DNS Suffix..:
   Link-local IPv6 Address.........: FE80::207:ECFF:FE64:22C3
   IPv6 Address....................: ::
   IPv4 Address....................: 192.168.1.20
   Subnet Mask.....................: 255.255.255.0
   Default Gateway.................: ::
                                     0.0.0.0

Bluetooth Connection:

   Connection-specific DNS Suffix..:
   Link-local IPv6 Address.........: ::
   IPv6 Address....................: ::
   IPv4 Address....................: 0.0.0.0
   Subnet Mask.....................: 0.0.0.0
   Default Gateway.................: ::
                                     0.0.0.0
```

ping

The "ping" command ping command allows you to send a signal to another device, and if that device is active, it will send a response back to the sender. The "ping" command is a subset of the ICMP (Internet Control Message Protocol), and it uses what is called an "echo request". So, when you ping a device you send out an echo request, and if the device you pinged is active or online, you get an echo response.

```
Packet Tracer PC Command Line 1.0
C:\>ping 192.168.1.22

Pinging 192.168.1.22 with 32 bytes of data:

Reply from 192.168.1.22: bytes=32 time<1ms TTL=128
Reply from 192.168.1.22: bytes=32 time<1ms TTL=128
Reply from 192.168.1.22: bytes=32 time<1ms TTL=128
Reply from 192.168.1.22: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.22:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

tracert

This command lets you see all steps a packet takes to the destination. For example, if we send a packet to www.google.com, it actually goes through a couple of routers to reach the destination. The packet will first go to your router, and then it will go to all kinds of different routers before it reaches Google servers. We can also use the term "hops" instead of routers. Let's run the command and see what kind of results we get.

```
C:\>tracert 192.168.1.22

Tracing route to 192.168.1.22 over a maximum of 30 hops:

  1   0 ms      0 ms      0 ms      192.168.1.22

Trace complete.
```

nslookup

The nslookup command will fetch the DNS records for a given domain name or an IP address. Remember the IP addresses and domain names are stored in DNS servers, so the nslookup command lets you query the DNS records to gather information.

Command Prompt - nslookup

```
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>nslookup
Default Server:  UnKnown
Address:  192.168.1.1

> www.google.com
Server:  UnKnown
Address:  192.168.1.1

Non-authoritative answer:
Name:    www.google.com
Addresses:  2404:6800:4009:828::2004
          142.250.192.36

>
```

Experiment 10

10.Use of NS2,NS3,Omnet and other simulators used in networking.

Use of NS2, NS3,Omnet and other simulators used in internetworking

1.NS2 stands for Network Simulator Version 2. It is an open-source event-driven simulator designed specifically for research in computer communication networks.

Features of NS2

1. It is a discrete event simulator for networking research.

2. It provides substantial support to simulate bunch of protocols like TCP, FTP, UDP, https and DSR.

3. It simulates wired and wireless network.

4. It is primarily Unix based.

5. Uses TCL as its scripting language.

6. Otcl: Object oriented support

7. Tclcl: C++ and otcl linkage

8. Discrete event scheduler

Basic Architecture

NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events. The C++ and the OTcl are linked together using TclCL

2. NS3 is a discrete event network simulator for Internet. NS3 helps to create various virtual nodes (i.e., computers in real life) and with the help of various Helper classes it allows us to install devices, internet stacks, application, etc to our nodes.

Using NS3 we can create PointToPoint, Wireless, CSMA, etc connections between nodes. PointToPoint connection is same as a LAN connected between two computers. Wireless connection is same as WiFi connection between various computers and routers. CSMA connection is same as bus topology between computers. After building connections we try to install NIC to every node to enable network connectivity.

When network cards are enabled in the devices, we add different parameters in the channels (i.e., real world path used to send data) which are data-rate, packet size, etc. Now we use Application to generate traffic and send the packets using these applications.

Ns3 gives us special features which can be used for real life integrations. Some of these features are:

1. Tracing of the nodes:

NS3 allows us to trace the routes of the nodes which helps us to know how much data is send or received. Trace files are generated to monitor these activities.

2. NetAnim:

It stands for Network Animator.It is an animated version of how network will look in real and how data will be transferred from one node to other.

3. Pcap file:

NS3 helps to generate pcap file which can be used to get all information of the packets (e.g., Sequence number, Source IP, destination IP, etc). These pcaps can be seen using a software tool known as wireshark.

4. gnuPlot:

GnuPlot is used to plot graphs from the data which we get from trace file of NS3. Gnuplot gives more accurate graph compare to other graph making tools and also it is less complex than other tools.

3. OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators. “Network” is meant in a broader sense that includes wired and wireless communication networks, on-chip networks, queueing networks, and so on. Domain-specific functionality such as support for sensor networks, wireless ad-hoc networks, Internet protocols, performance modeling, photonic networks, etc., is provided by model frameworks, developed as independent projects. OMNeT++ offers an Eclipse-based IDE, a graphical runtime environment, and a host of other tools. There are extensions for real-time simulation, network emulation, database integration, SystemC integration, and several other functions. OMNeT++ is distributed under the Academic Public License.

Although OMNeT++ is not a network simulator itself, it has gained widespread popularity as a network simulation platform in the scientific community as well as in industrial settings, and building up a large user community.

OMNeT++ provides a component architecture for models. Components (modules) are programmed in C++, then assembled into larger components and models using a high-level language (NED). Reusability of models comes for free. OMNeT++ has extensive GUI support, and due to its modular architecture, the simulation kernel (and models) can be embedded easily into your applications.

The main ingredients of OMNeT++ are:

- Simulation kernel library (C++)
- The NED topology description language
- Simulation IDE based on the Eclipse platform
- Interactive simulation runtime GUI (Qtenv)
- Command-line interface for simulation execution (Cmdenv)
- Utilities (makefile creation tool, etc.)
- Documentation, sample simulations, etc.