

Experiment No: 01

Experiment Name:

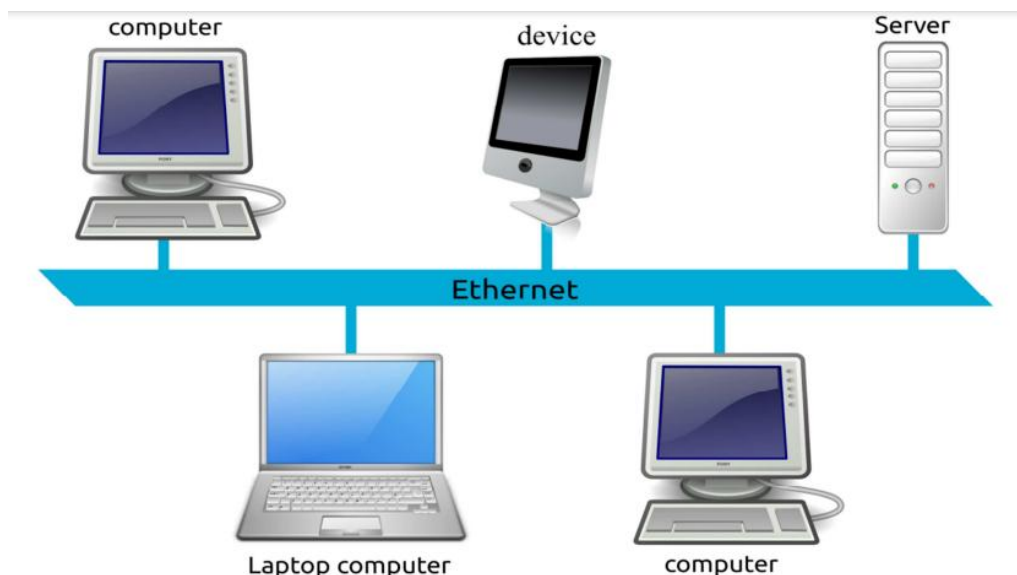
Building LAN Network using Hub

Object:

Build a LAN network using Ethernet cable

Theory:

LAN is a computer network that connects a relatively small area (a single building or group of buildings). Most LANs connect workstations and computers to each other. Each computer (also known as a “node”), has its own processing unit and executes its own programs; however, it can also access data and devices anywhere on the LAN.



Component:

1. Two or more computers/Laptops. (Example: 5 Computers).
2. Two or more NIC (Network Interface Card). In this case 5 NICs for 5 computers.

3. LAN cable (UTP cables with the desired length with RJ 45).
4. Eight port Hub.

Procedure:

1. Install the NIC into the PCMCIA slot of the PC's mother board. NIC is a hardware that interfaces between the computer and the network. The NIC has RJ 45 type port.
2. Take the desired length of 5 LAN cables and terminate both the ends with a RJ 45 type connector.
3. Now, connect one end of each of 5 cables to the computer and the other end to the Hub.
4. Now, The physical network is ready
5. Assign IP address, Subnet Mask to every PC.
6. To verify the IP address and Subnet Mask assignment in each computer, go to command prompt and type `CD:\>ipconfig /all`
7. To check the connectivity between PCs, go to command prompt and type `CD:\>ping`
8. In return to the ping, the computer receives a reply back as an acknowledgement, thus the LAN connectivity is UP.

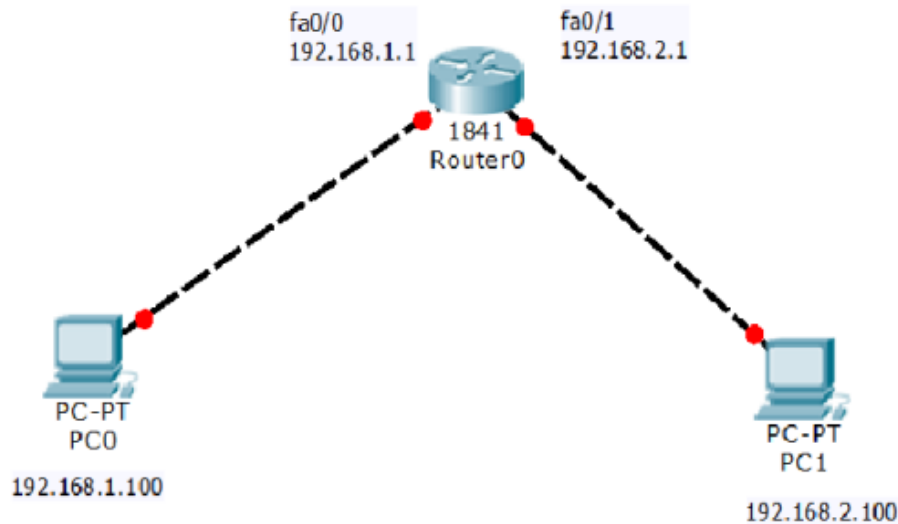
Discussion:

1. What is the difference between a physical design and a logical design?
2. What is the difference between Fast Ethernet and regular Ethernet?
3. List the advantages and disadvantages of local area networks.
4. What are the primary differences between baseband technology and broadband technology?

Experiment No: 02

Experiment Name:

Design and configure a network infrastructure with two networks connected by a Cisco router. Also configure DHCP for one network.



Tools required:

- CISCO Packet tracer

Objective of the Experiment:

After completing this Lab student should able to:

- Understand client server model practically.
- Can able to construct client server model.
- Can able configure DHCP server.
- Can able to configure HTTP server.
- Can be able to construct wireless topologies using switches and Access points.

Walk-through Tasks:

1. Create a client server model by using one switch and three computers assign IPaddresses, subnet masks to server and also to PC's of the network. Enable server to provideHTTP services, access HTTP server from any one of the clients.

For achieving this you have to follow following steps:

- Open Cisco packet tracer.
- Add one switch and three computers in work space of cisco packet tracer.
- Add a server in cisco packet tracer workspace, now you will see, your workspace in this form.
- Connect server with switch by selecting copper wire and in the same way, connect computers with switch by selecting copper wire:
- Assign IP address and subnet mask to server, just like we do in PC's, click on the server, go in Desktop tab and in IP configuration application, give IP addresses and subnet masks as follow:
- Give IP addresss in IP confriguration.
- In the same way assign, IP addresses and subnet masks in all PC's.
- For accessing the HTTP services of the server, first go in the server and see that is, the HTTP service is on or off. For checking this click on the server and go in the services tab and make sure HTTP service is on:
- Here in this window above, you can see that we can add new file in it, it will act like as of apache server do, when we access that, so from new file we can add any new file in it (rather an image or textfile).
- Now to access this server go in the workspace and click on any one of the client, and go in the application of web browser, where you have to give IP address of the server and with slash (/give filename), give file name you want to access as shown in the figure:

2. Configuring DHCP server: For configuring DHCP server, do these steps:

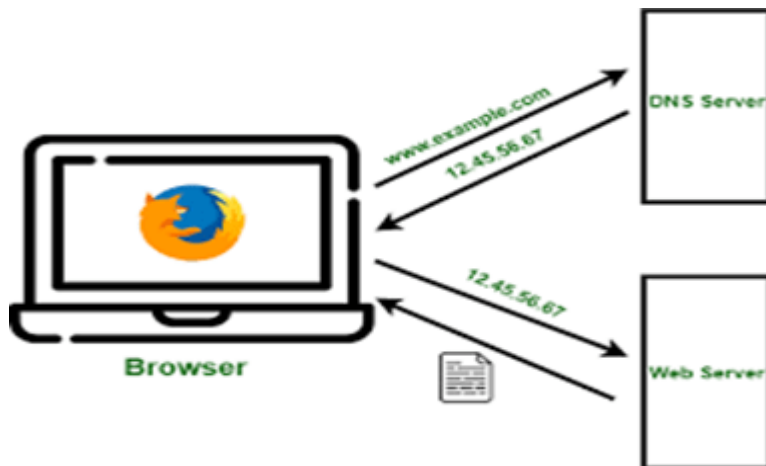
- Click on the server.
- Go in the services tab.
- Select DHCP service in the server terminal
- Click on the DHCP service and when new windows appear, click the check box to on the DHCP
- services of the server, and click on the save button to save the settings

Experiment No: 03

Experiment Name:

Install and configure DNS server using Linux platform.

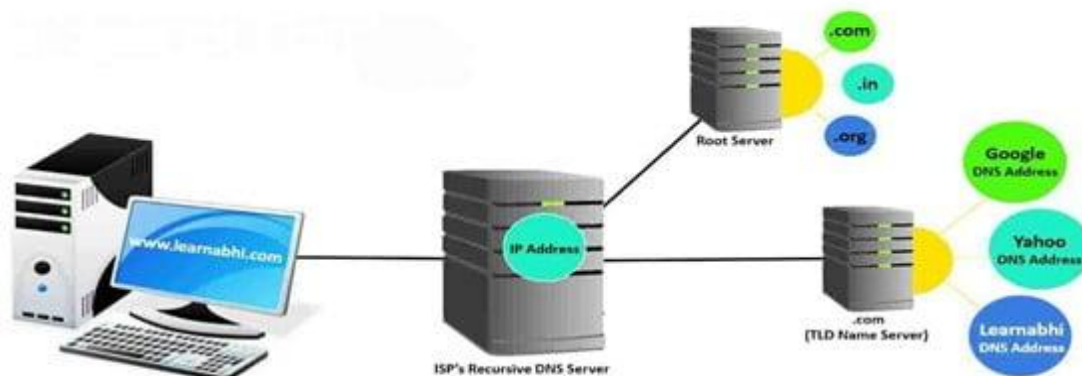
What is DNS?



The Domain Name System (DNS) turns domain names into IP addresses, which browsers use to load internet pages. Every device connected to the internet has its own IP address, which is used by other devices to locate the device.

How DNS works?

In the first step DNS resolver, also called recursive resolver designed to receive DNS queries from a web browser and other applications. The resolver receives a hostname, for example, **www.zindagi.com**, and is responsible for tracking down the IP addresses for that hostname.



After that root server is translating human-readable hostname into IP addresses. There are 13 logical root servers worldwide, denoted by letter A through M controlled by Cogen, the University of Maryland, and the U.S arm lab. The TLD nameserver takes the domain provided in the query and provides the IP of an authoritative name server. The authoritative name server takes the domain name and subdomain, and it returns the correct IP address to the DNS resolver.

9 Steps Of DNS Configuration

Let's see how to configure a DNS server in 9 easy steps.

Step 1

Install bind packages with their dependencies. BIND stands for Berkley Internet Naming Daemon. BIND is the most common program used for maintaining a name server on Linux.

```
# Yum install -y bind*
```

Step 2

Assign a static IP address in Linux OS. Configure network file.

```
# Vim /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
NAME=ens192
IPADDR=192.168.1.189
NETMASK=255.255.255.0
UUID=a4609655-d758-4897-99a9-d1f589cf4865
DEVICE=ens192
ONBOOT=no
~
~
~
```

Step 3

Add a host entry in the host file. Just add your local machine IP address with the domain name of the website which you are hosting on a local server. After the edit, the host file restarts the network and ping the specified IP address to see whether the data is exchanged or not.

```
# vim /etc/hosts
```

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.189 www.zindagi.com
~
~
~
~
~
~
~
```

Step 4

Add server IP to the resolv.conf file.

```
# Vim /etc/resolv.conf
```

```
# Generated by NetworkManager
search zindagi.com
nameserver 192.168.1.2
nameserver 8.8.8.8
~
~
~
~
```

Step 5

Configure /etc/named.conf file and edit the following lines.

```
# Vim /etc/named.conf
```

Step 6

Open /etc/named.rfc1912.zones file. Next, we need to add zone records for the forward zone file and reverse zone file. In this file, we set up a master forward record and master reverse DNS record.

Step 8

Change the group ownership for forward zone and reverse zone files.

```
# chgrp named /var/named/forward. Zone  
  
# chgrp named /var/named/reverse. Zone
```

Step 9

Restart the DNS service.

```
# Service named restart
```

In the next step, you should use the ns lookup utility to query the IP using the hostname and vice versa. You can also use dig to test DNS servers.

Experiment No: 04

Experiment Name:

Install and configure HTTP server using Linux platform.

Install Apache server

1. Run the following command to install Apache.

```
2. # apt-get install apache2
```

3. Run the following command to start the Apache process.

```
4. # /etc/init.d/apache2 start
```

5. Verify that the service is running by executing the following command.

```
6. # /etc/init.d/apache2 status
```

7. Run the following command to restart Apache.

```
8. # /etc/init.d/apache2 restart
```

Configure Apache server

The next step is to set up the web server configuration for the domain. The Apache configuration directory is `/etc/apache2` and `apache2.conf` is main Apache configuration file. Each domain needs its own Virtual Host configuration file. The configuration files use the `.conf extension`, and need to be saved in the `/etc/apache2/sites-available/` directory.

1. Create a file at `/etc/apache2/sites-available/yourdomain.com.conf` and add the following lines to it.

```
2. # nano /etc/apache2/sites-  
   available/yourdomain.com.conf  
3.  
4. <virtualhost *:80="">  
5. ServerAdmin webmaster@localhost  
6. ServerName yourdomain.com  
7. ServerAlias www.yourdomain.com  
8. DocumentRoot /var/www/yourdomain.com  
9. ErrorLog ${APACHE_LOG_DIR}/error.log  
10.     CustomLog ${APACHE_LOG_DIR}/access.log  
    combined  
11.     </virtualhost>
```

12. Create a directory for the website and then create `index.html` file for the website.

```
13.     # mkdir /var/www/yourdomain.com
```

14. Add some content to `index.html`.

```
15.     # vi /var/www/yourdomain.com/index.html
```

16. Restart Apache service for the above changes to take effect.

```
17.     # /etc/init.d/apache2 restart  
18.     or  
19.     # sudo systemctl restart apache2
```


20. Open any browser and enter the website URL.

```
21.     http://yourdomain.com
```


Test Apache server

The Apache web server can be tested by entering the server IP address into the browser's address bar:

`http://your_server_ip`



Apache2 Ubuntu Default Page



It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in** [`/usr/share/doc/apache2/README.Debian.gz`](#). Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.