

Computer Networks - Lab 05

Objective

The objective of this lab is to provide students with an understanding of the Application Layer and its protocols. Specifically, the lab focuses on DNS (Domain Name System) server configuration using Packet Tracer. Students will gain practical experience in configuring Web Server, and DNS server, understanding its role in translating domain names to IP addresses, and exploring the functionality of DNS protocols.

Learning Outcomes

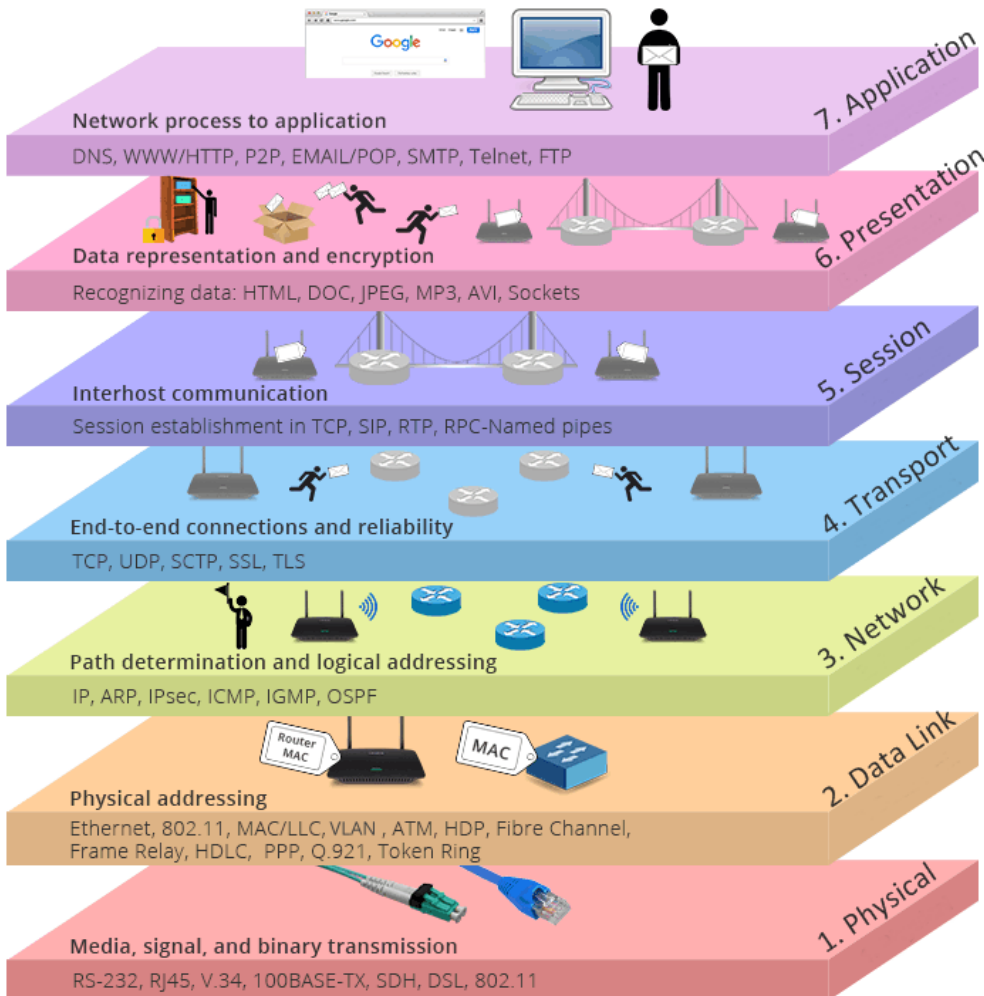
By the end of this lab, students will be able to:

- Understand the Application Layer in the TCP/IP protocol stack and its role in supporting network applications.
- Configure a Web server in Packet Tracer
- Configure a DNS server in Packet Tracer to host a domain and provide domain name resolution services.
- Understand the process of domain name resolution and the interaction between DNS clients and servers.
- Troubleshoot common DNS configuration issues and perform basic DNS testing using commands like nslookup

Table of Contents

Objective.....	1
Learning Outcomes.....	1
OSI Model	3
Protocols of Application Layer	3
Server.....	4
Website Server.....	6
DNS Server	7
DNS server configuration in Packet Tracer	9
DHCP, DNS and Web Server configuration in cisco packet tracer	11
Lab Tasks.....	18

OSI Model



Protocols of Application Layer

The application layer provides several protocols which allow any software to easily send and receive information and present meaningful data to its users.

The following are some of the protocols which are provided by the application layer.

TELNET: Telnet stands for Telecommunications Network. This protocol is used for managing files over the Internet. It allows the Telnet clients to access the resources of Telnet server. Telnet uses port number 23.

DNS: DNS stands for Domain Name System. The DNS service translates the domain name (selected by user) into the corresponding IP address. For example- If you choose the domain name as www.abcd.com, then DNS must translate it as 192.36.20.8 (random IP address written just for understanding purposes). DNS protocol uses the port number 53.

DHCP: DHCP stands for Dynamic Host Configuration Protocol. It provides IP addresses to hosts. Whenever a host tries to register for an IP address with the DHCP server, DHCP server provides lots of information to the corresponding host. DHCP uses port numbers 67 and 68.

FTP: FTP stands for File Transfer Protocol. This protocol helps to transfer different files from one device to another. FTP promotes sharing of files via remote computer devices with reliable, efficient data transfer. FTP uses port number 20 for data access and port number 21 for data control.

SMTP: SMTP stands for Simple Mail Transfer Protocol. It is used to transfer electronic mail from one user to another user. SMTP is used by end users to send emails with ease. SMTP uses port numbers 25 and 587.

HTTP: HTTP stands for Hyper Text Transfer Protocol. It is the foundation of the World Wide Web (WWW). HTTP works on the client server model. This protocol is used for transmitting hypermedia documents like HTML. This protocol was designed particularly for the communications between the web browsers and web servers, but this protocol can also be used for several other purposes. HTTP is a stateless protocol (network protocol in which a client sends requests to server and server responses back as per the given state), which means the server is not responsible for maintaining the previous client's requests. HTTP uses port number 80.

NFS: NFS stands for Network File System. This protocol allows remote hosts to mount files over a network and interact with those file systems as though they are mounted locally. NFS uses the port number 2049.

SNMP: SNMP stands for Simple Network Management Protocol. This protocol gathers data by polling the devices from the network to the management station at fixed or random intervals, requiring them to disclose certain information. SNMP uses port numbers 161 (TCP) and 162 (UDP).

Server

A server is a computer program or device that provides a service to another computer program and its user, also known as the client. In a data center, the physical computer that a server program runs on is also frequently referred to as a server. That machine might be a dedicated server or it might be used for other purposes.

Types of servers: Servers are often categorized in terms of their purpose. A few examples of the types of servers available are as follows:

Web server: a computer program that serves requested HTML pages or files. In this case, a web browser acts as the client.

Application server: a program in a computer in a distributed network that provides the business logic for an application program.

Proxy server: software that acts as an intermediary between an endpoint device, such as a computer, and another server from which a user or client is requesting a service.

Mail server: an application that receives incoming emails from local users -- people within the same domain -- and remote senders and forwards outgoing emails for delivery.

Virtual server: a program running on a shared server that is configured in such a way that it seems to each user that they have complete control of a server.

Blade server: a server chassis housing multiple thin, modular electronic circuit boards, known as server blades. Each blade is a server in its own right, often dedicated to a single application

File server: a computer responsible for the central storage and management of data files so that other computers on the same network can access them.

Policy server: a security component of a policy-based network that provides authorization services and facilitates tracking and control of files.

Database server: this server is responsible for hosting one or more databases. Client applications perform database queries that retrieve data from or write data to the database that is hosted on the server.

Print server: this server provides users with access to one or more network-attached printers -- or print devices as some server vendors call them. The print server acts as a queue for the print jobs that users submit. Some print servers can prioritize the jobs in the print queue based on the job type or on who submitted the print job.

Server components

Hardware

Servers are made up of several different components and subcomponents. At the hardware level, servers are typically made up of a rack mount chassis containing a power supply, a system board, one or more CPUs, memory, storage, a network interface and a power supply.

Most server hardware supports out-of-band management through a dedicated network port. Out-of-band management enables low-level management and monitoring of the server, independently of the operating system. Out-of-band management systems can be used to remotely power the server on or off, to install an operating system, and to perform health monitoring.

Operating systems

Another component is the server operating system. A server operating system, such as Windows Server or Linux, acts as the platform that enables applications to run. The operating system provides applications access to the hardware resources that they need and enables network connectivity.

The application is what enables the server to do its job. For example, a database server would run a database application. Likewise, an email server would need to run a mail application.

\

Website Server.

A web server powers the site. This genre of server focuses on serving web content to clients.

Web servers simply take “GET” and “POST” requests from clients (among other verbs). A “GET” request is when a client simply wants to retrieve information and doesn’t have any information to submit to the server.

A “POST” request on the other hand is when a client *does* have information to share with the server and expects a response back. For example, filling up a form on a web server and clicking the submit button is a “POST” request from the client to the server.

Web servers are typically “headless” in nature. This is to preserve the memory on the server and ensure that there’s enough to power the operating system and applications on the server.

“Headless” means that it doesn’t run like a traditional home computer, but rather just serves content. The administrators of these servers can only connect to them through command line terminals.

Remember that these types of servers can run any type of application just like your home computer can.

They can also run on any operating system, as long as they obey the general “rules” of the web.

Modern web applications usually run on a series of layers, starting with server-side scripts and programs that process data (e.g PHP, ASP.NET etc), and ending with client-side scripting (e.g Javascript) that programs how the data should be displayed.

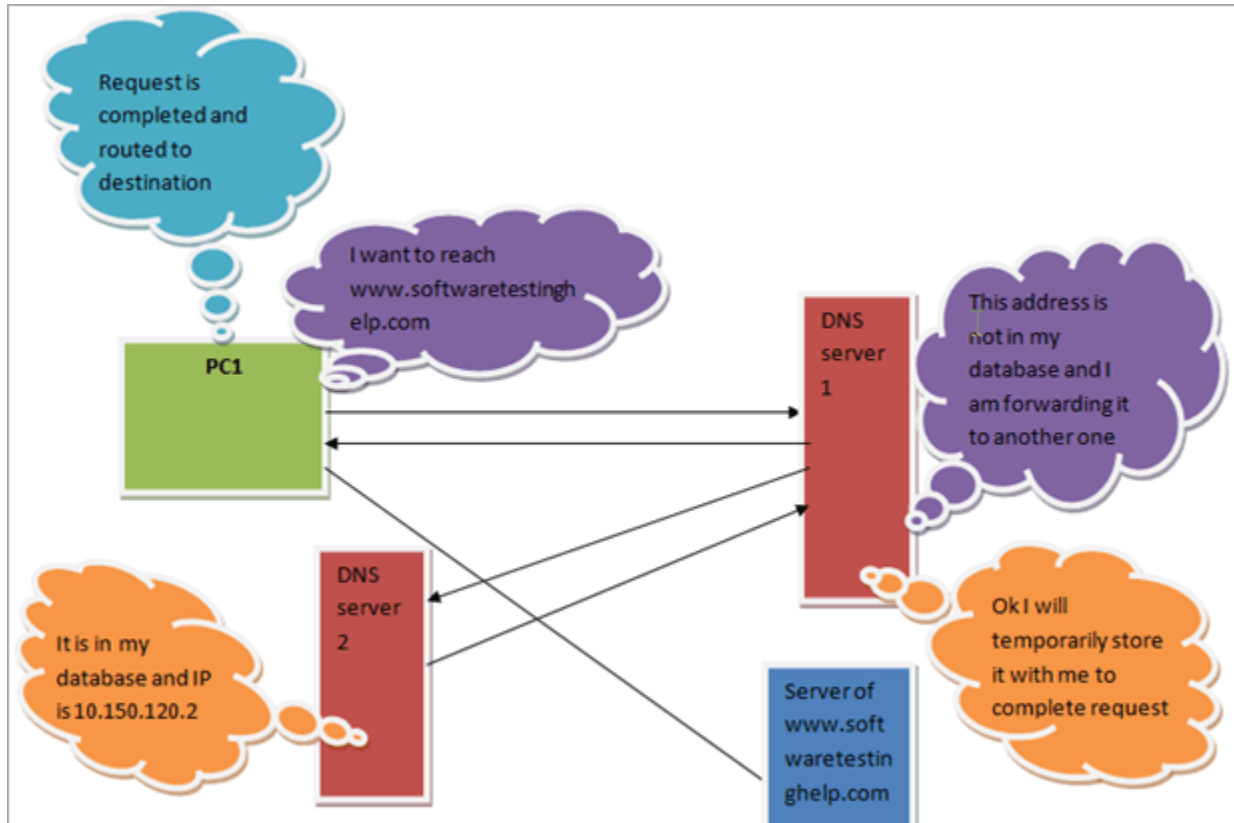
A web browser then renders the content accordingly to show the page as you’re reading it now.

Some popular web servers include Microsoft IIS, Apache, Nginx etc.

Some Ports used for Webservers: Port 80 for HTTP (not encrypted) and Port 443 for HTTPS (encrypted).

DNS Server

DNS Server is needed for resolving hostnames to their IP addresses. Normally your ISP will provide you with DNS Service. You may have your own DNS Server, which will resolve hostnames by forwarding them to ISP's DNS Server and cache the addresses also.



As shown in the above figure, when we request for a web page from our PC on the Internet like PC1 is requesting for www.softwaretestinghelp.com, then resolving the domain name query and providing the respective IP address in return is the part of work of the DNS server.

DNS server stores the database of all the relevant IP addresses mapped with their respective domain names

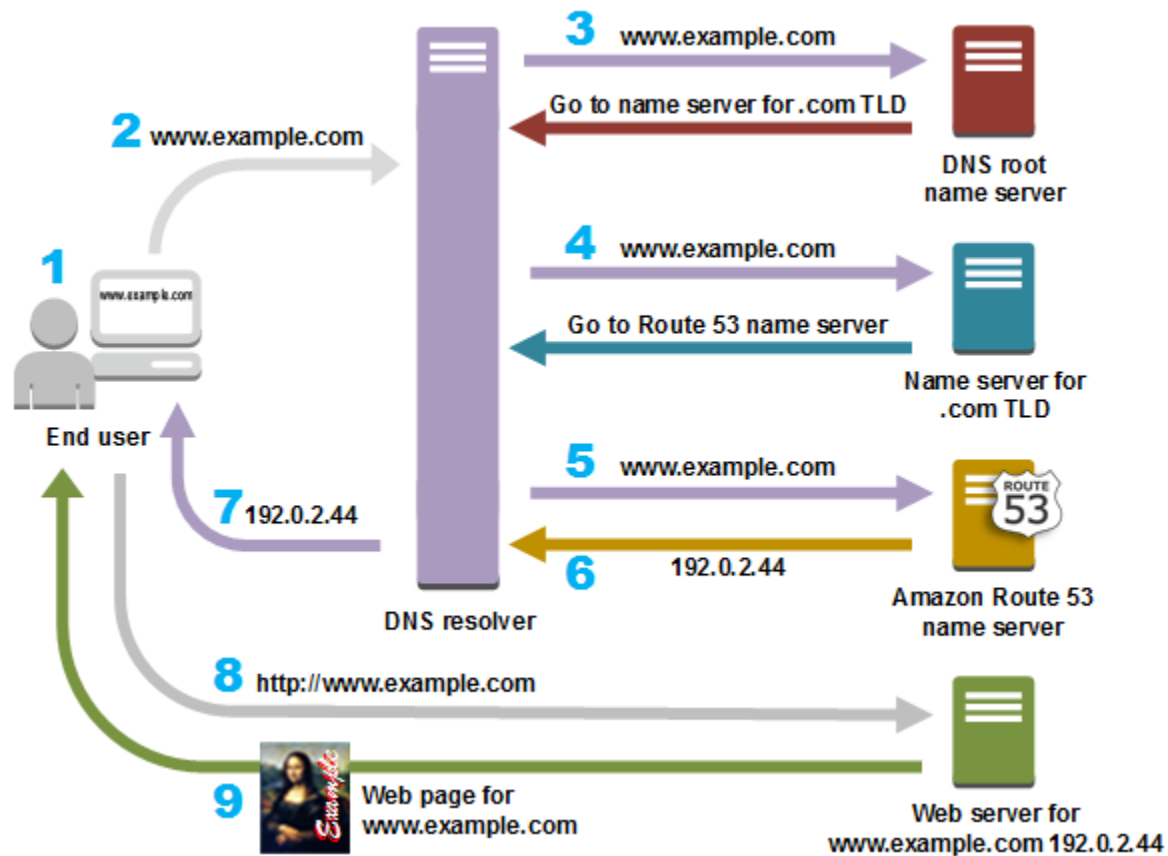
The DNS query for requesting the IP address in respect to the domain name goes to the DNS server 1 from PC1. The server checks within itself, if it has the IP address regarding the query, and it returns a DNS response with the resolution.

Otherwise, it forwards it to another DNS server 2 requesting for information. This time it gets the resolution from the DNS 2 and it gets mapped with the IP address i.e. 10.150.120.2 corresponding to the Domain name in response and sends it back to PC1.

The PC1 now have the destination IP address and it can communicate further with the known IP address as per the routing.

Now the question arises, as of how the PC will come to know which DNS should be used to get the IP address.

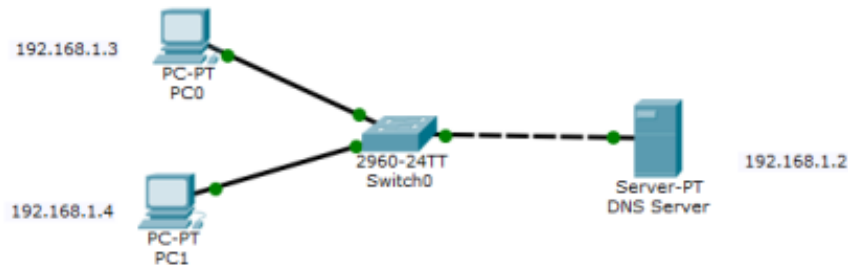
The answer to this is when we connect our system to the ISP, the network devices like a router or switch which assigns the routing information and other configurations as well send which or how many DNS server the PC should connect with to get the address translation.



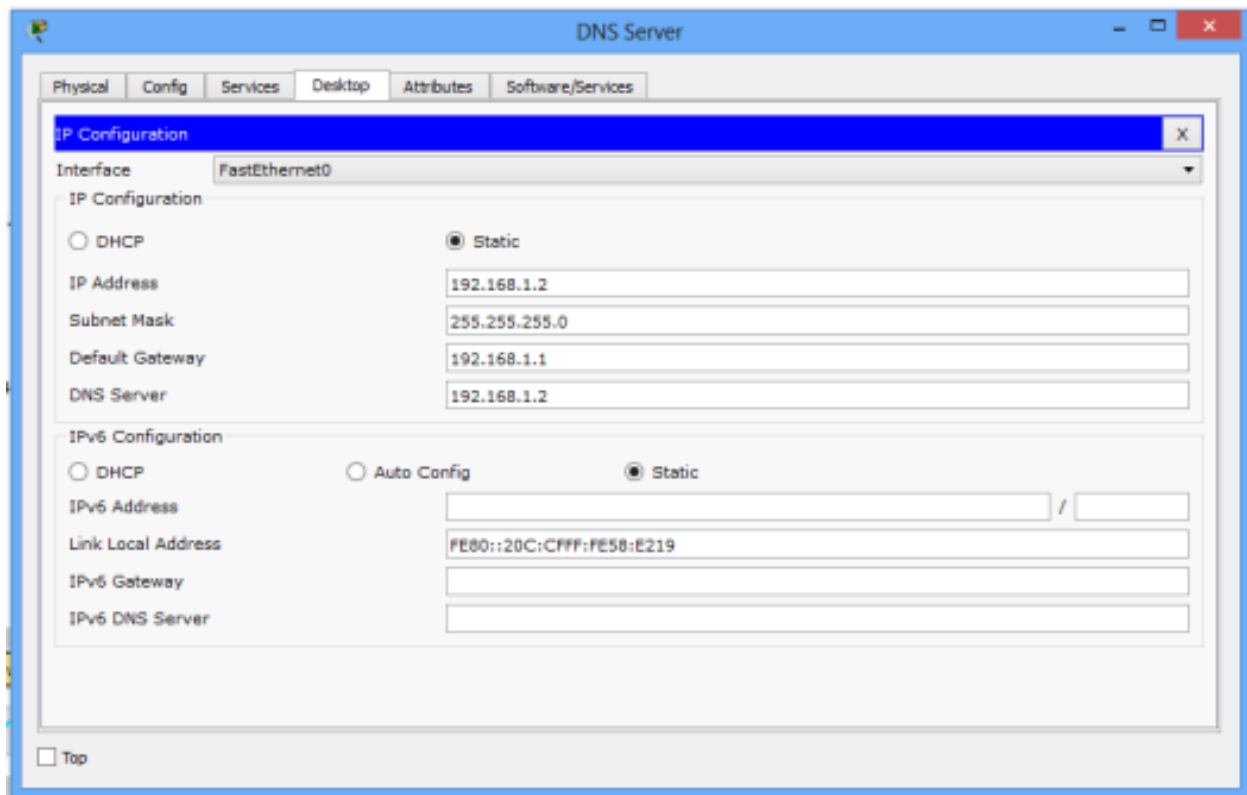
For more details: <https://aws.amazon.com/route53/what-is-dns/>

DNS server configuration in Packet Tracer

DNS server configuration in Packet Tracer



Configure static IP addresses on the PCs and the server. **Server: IP address:** 192.168.1.2 **Subnet mask:** 255.255.255.0 **Default gateway:** 192.168.1.1 **DNS Server:** 192.168.1.2



PC0 IP add: 192.168.1.3 **Subnet mask:** 255.255.255.0 **Default gateway:** 192.168.1.1 **DNS server:** 192.168.1.2

PC1 IP address: 192.168.1.4 **Subnet mask:** 255.255.255.0 **Default gateway:** 192.168.1.1 **DNS Server:** 192.168.1.2

Configure DNS service on the generic server.

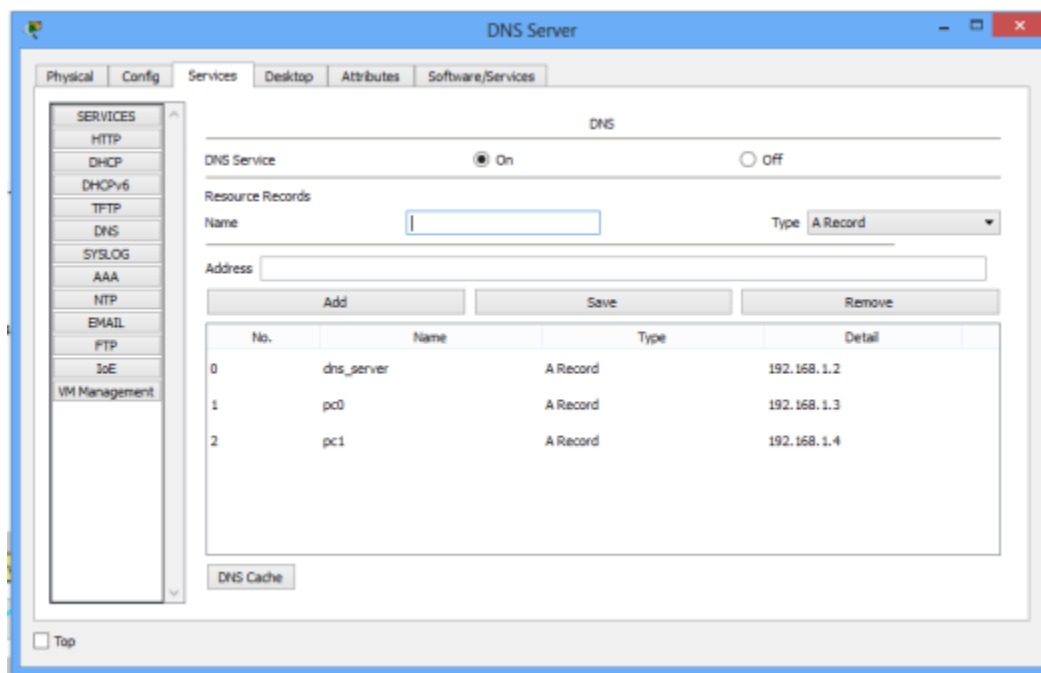
To do this, click on the server, then Click on **Services** tab. Click on **DNS server** from the menu. First turn **ON** the DNS service, then define **names** of the hosts and their corresponding **IP addresses**.

For example, to specify the DNS entry for PC0: In the **name** and **address** fields, type:

Name: PC0 **Address:** 192.168.1.3

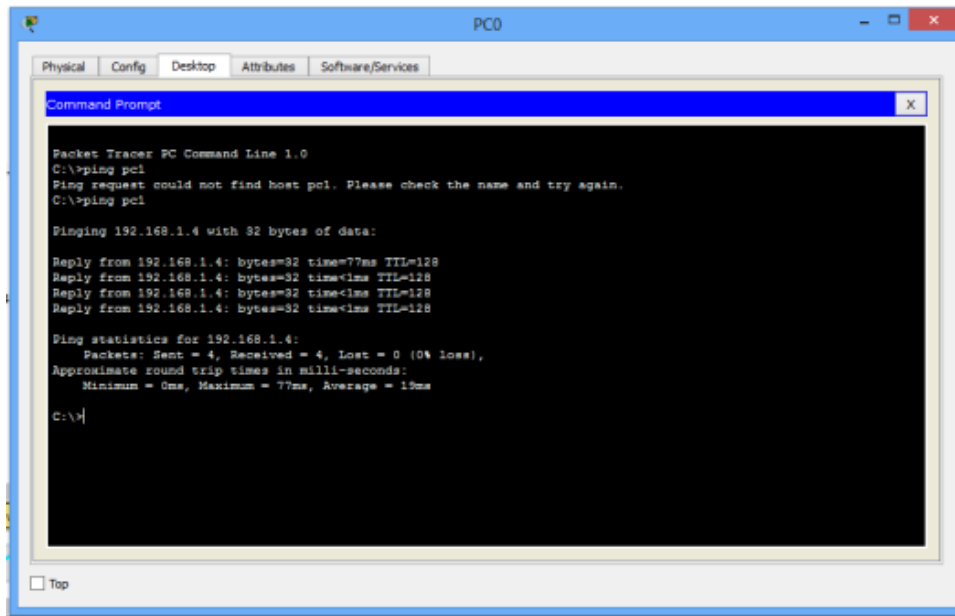
Click on **add** then **save**. Repeat this for the PC1 and the server.

Once you're done, your DNS entries will look like this:



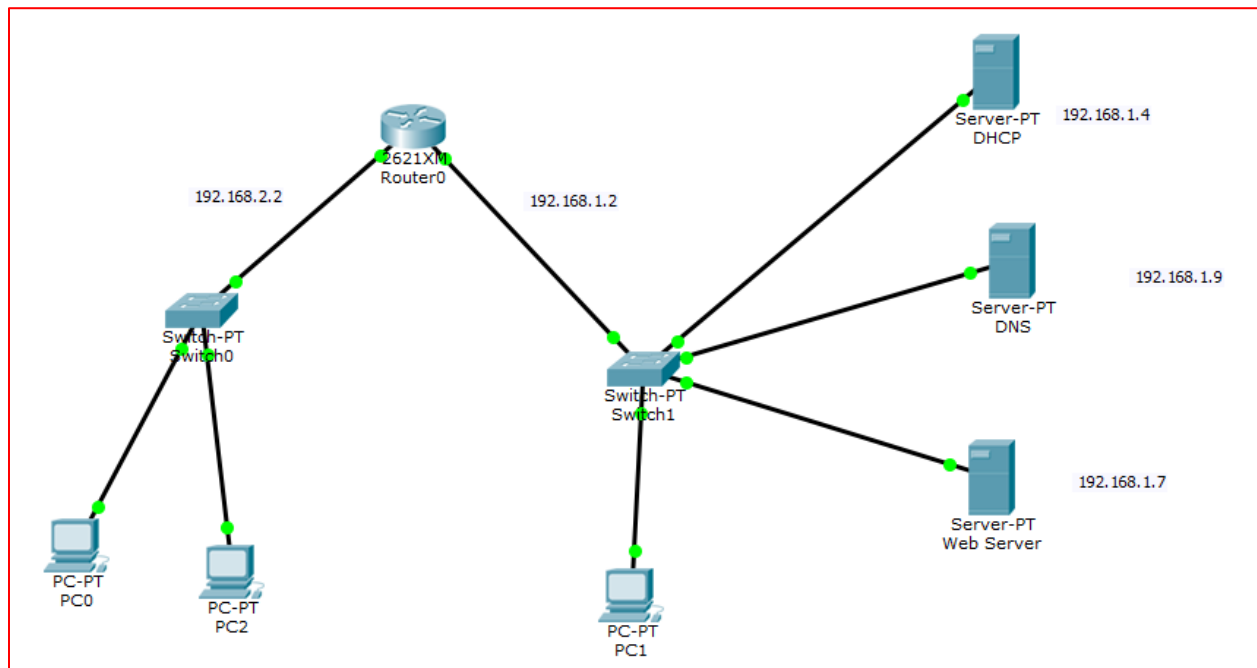
Finally, Test **domain name – IP resolution**. Ping the hosts from one another using their names instead of their IP addresses. If the DNS service is turned on and all IP configurations are okay, then ping should work.

For example, ping PC1 from PC0. Ping should be successful.



DHCP, DNS and Web Server configuration in cisco packet tracer

Build the network topology:



On the router, configure interface fa0/0 to act as the default gateway for our LAN.

Assign Ip address to F0/0 and F0/1 of router.

The screenshot shows the configuration window for the FastEthernet0/0 interface on Router0. The left sidebar has tabs for Physical, Config, and CLI. Under the Config tab, there are sections for GLOBAL (Settings, Algorithm Settings), ROUTING (Static, RIP), and INTERFACE (FastEthernet0/0, FastEthernet0/1). The FastEthernet0/0 interface is selected. The configuration fields are: Port Status (checked On), Bandwidth (radio buttons for 100 Mbps and 10 Mbps, with 100 Mbps selected and Auto checked), Duplex (radio buttons for Half Duplex and Full Duplex, with Full Duplex selected and Auto checked), MAC Address (00E0.8F82.5301), IP Configuration (IP Address: 192.168.1.2, Subnet Mask: 255.255.255.0), and Tx Ring Limit (10).

The screenshot shows the configuration window for the FastEthernet0/1 interface on Router0. The left sidebar has tabs for Physical, Config, and CLI. Under the Config tab, there are sections for GLOBAL (Settings, Algorithm Settings), ROUTING (Static, RIP), and INTERFACE (FastEthernet0/0, FastEthernet0/1). The FastEthernet0/1 interface is selected. The configuration fields are: Port Status (checked On), Bandwidth (radio buttons for 100 Mbps and 10 Mbps, with 100 Mbps selected and Auto checked), Duplex (radio buttons for Half Duplex and Full Duplex, with Full Duplex selected and Auto checked), MAC Address (00E0.8F82.5302), IP Configuration (IP Address: 192.168.2.2, Subnet Mask: 255.255.255.0), and Tx Ring Limit (10).

Apply follow commands in configuration mode on Router

```
ip dhcp pool P1
```

```
network 192.168.1.0 255.255.255.0
```

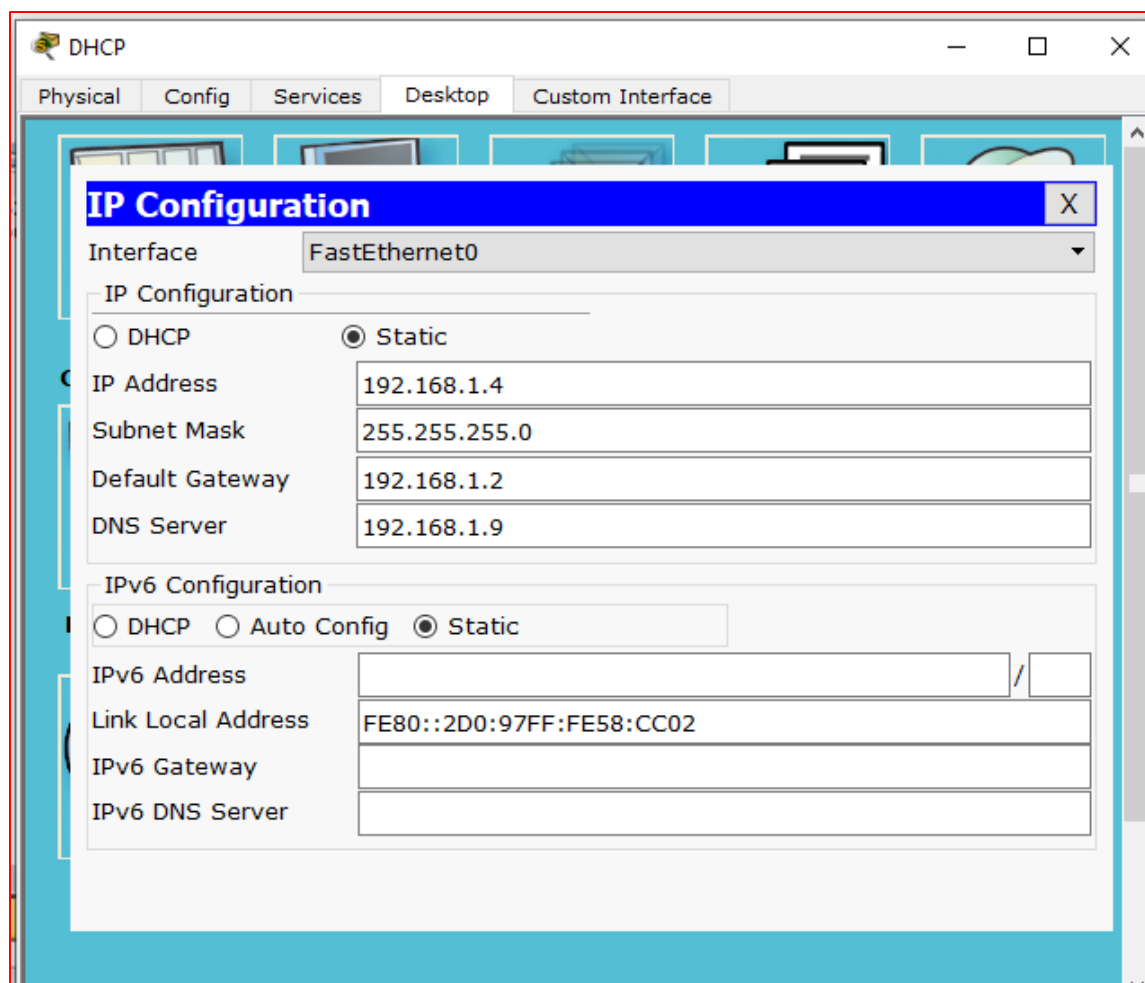
```
default-router 192.168.1.2
```

```
ip dhcp pool P2
```

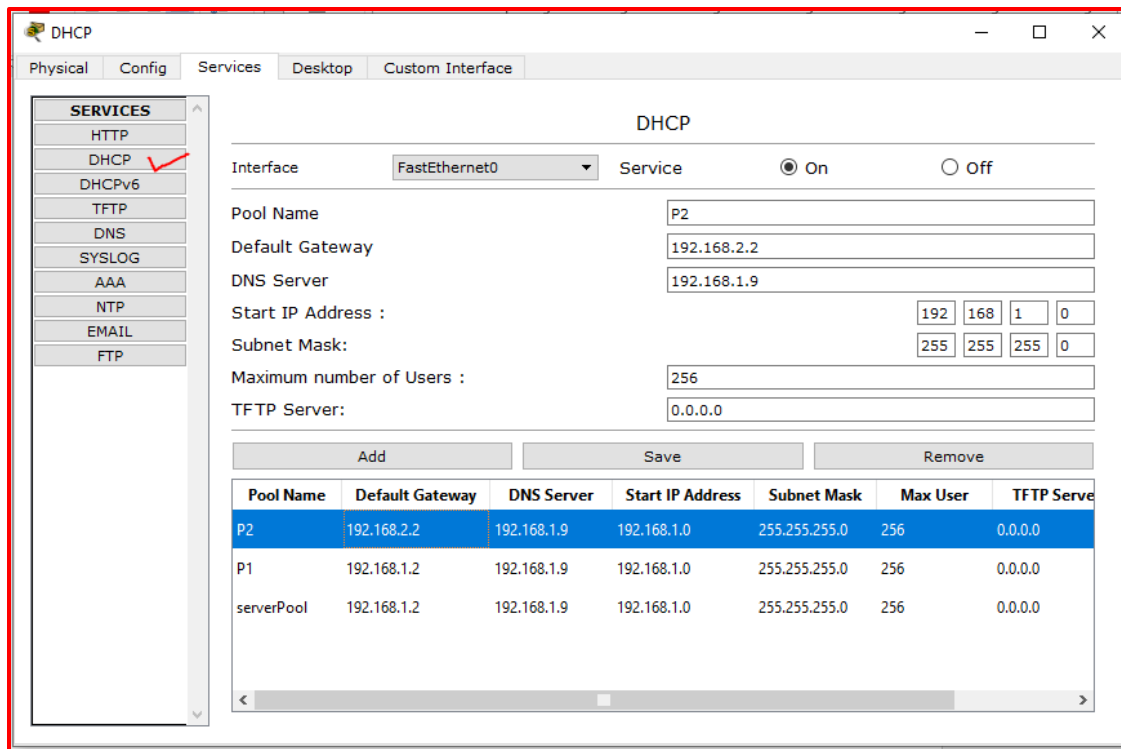
```
network 192.168.2.0 255.255.255.0
```

```
default-router 192.168.2.2
```

Apply follows setting on DHCP IP Configurations

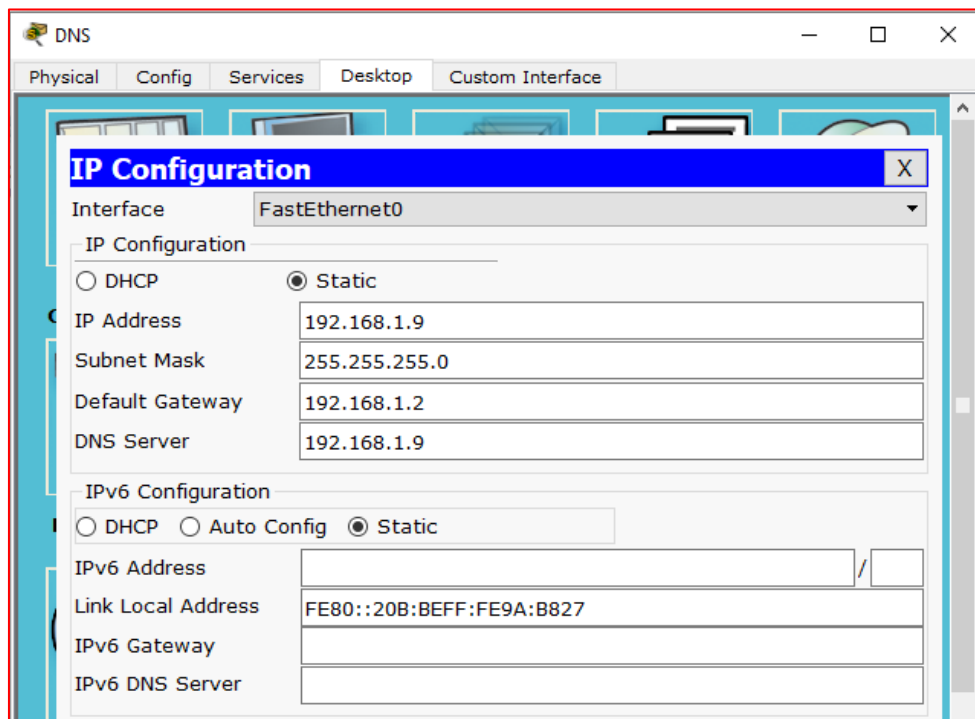


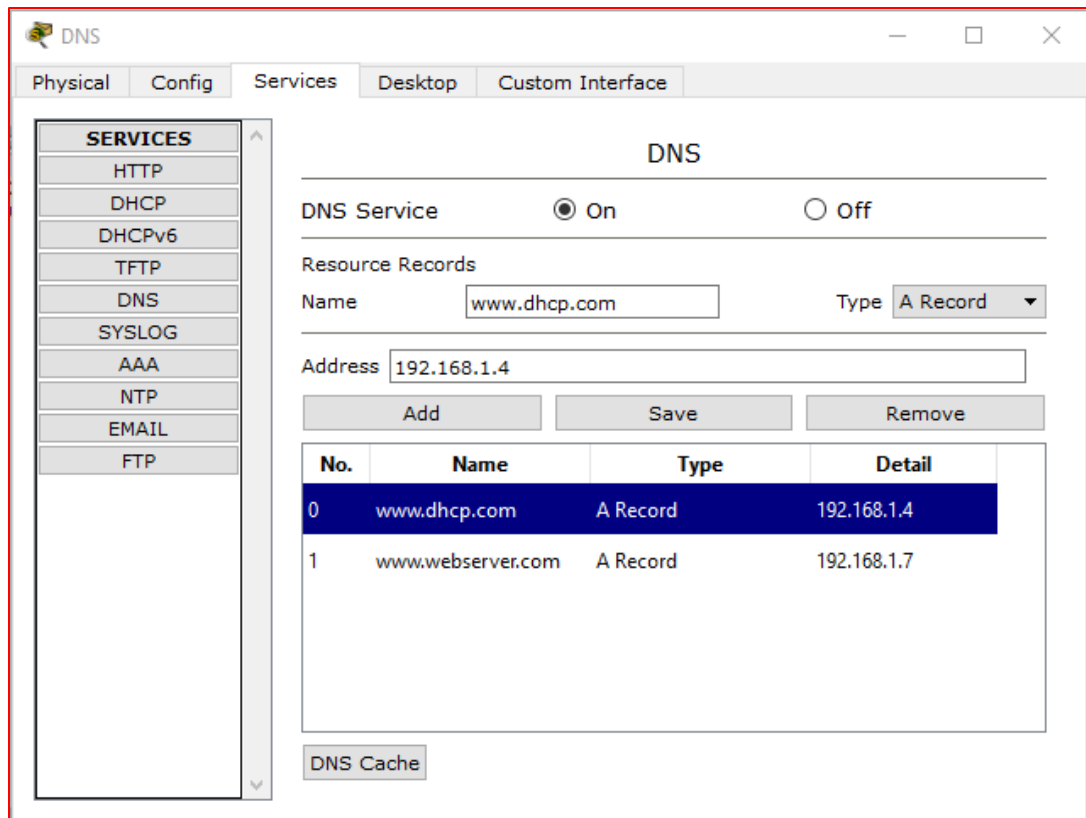
Enable DHCP Services and Add Pool P1 and Pool P2 with respective Ip Address



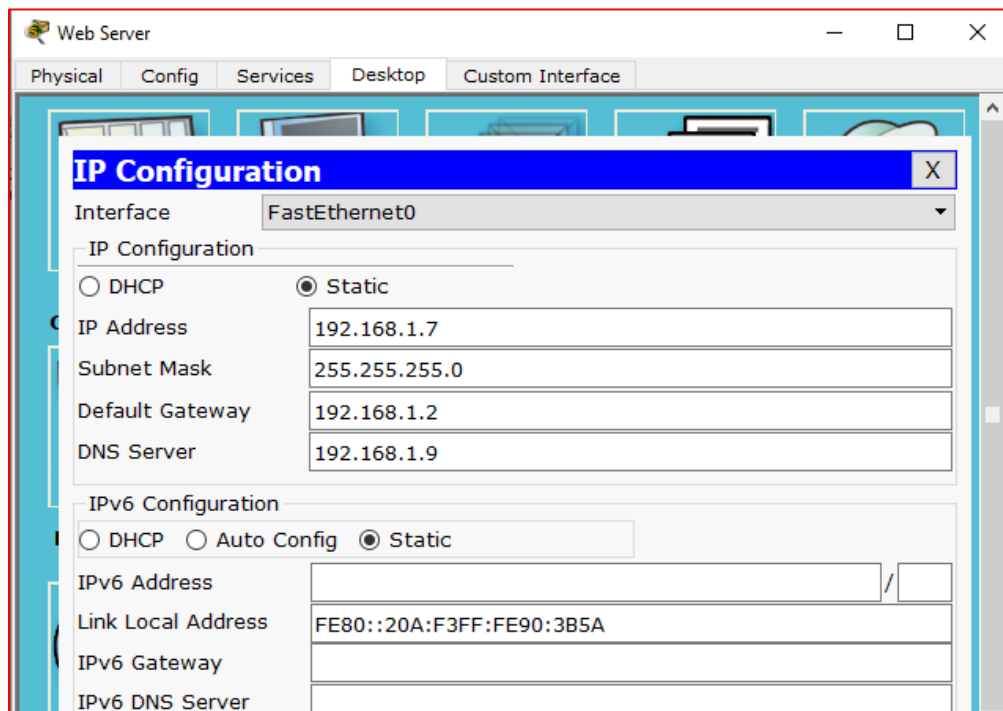
Apply follows setting on DNS IP Configurations

Enable DNS Services of DNS and add two resources records

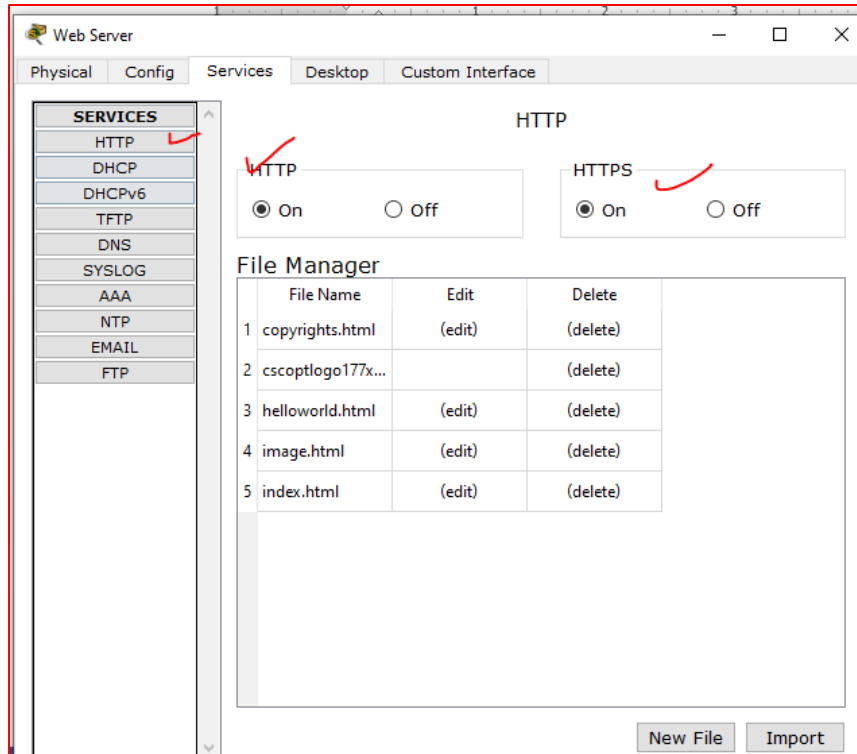




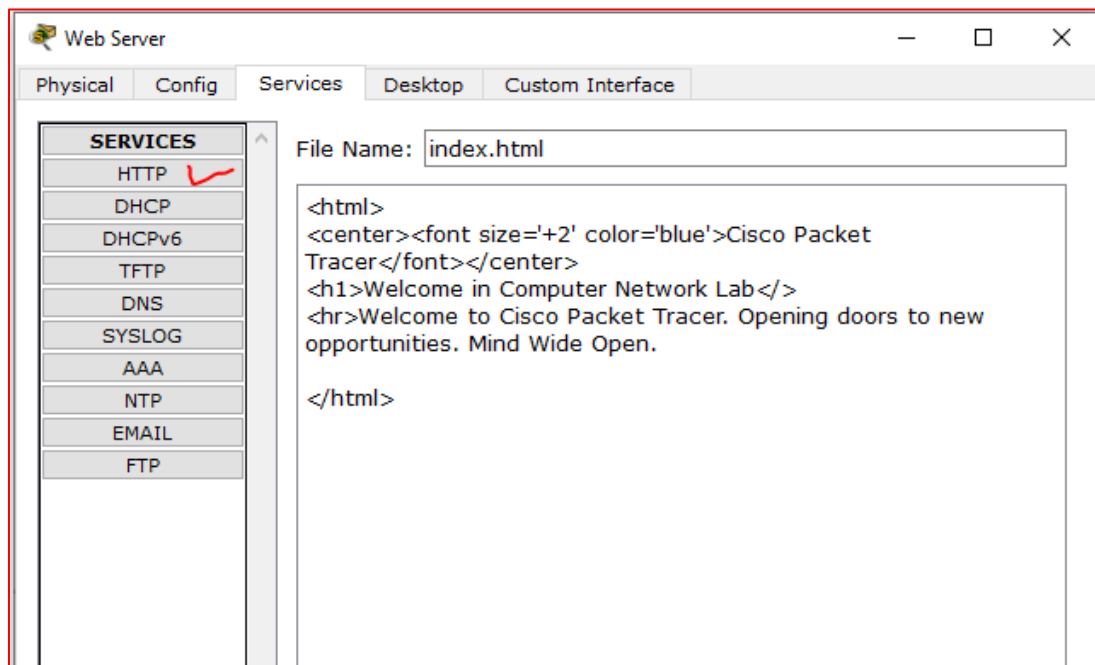
Apply follows setting on Web Server



Enable Http and Https in Web Server



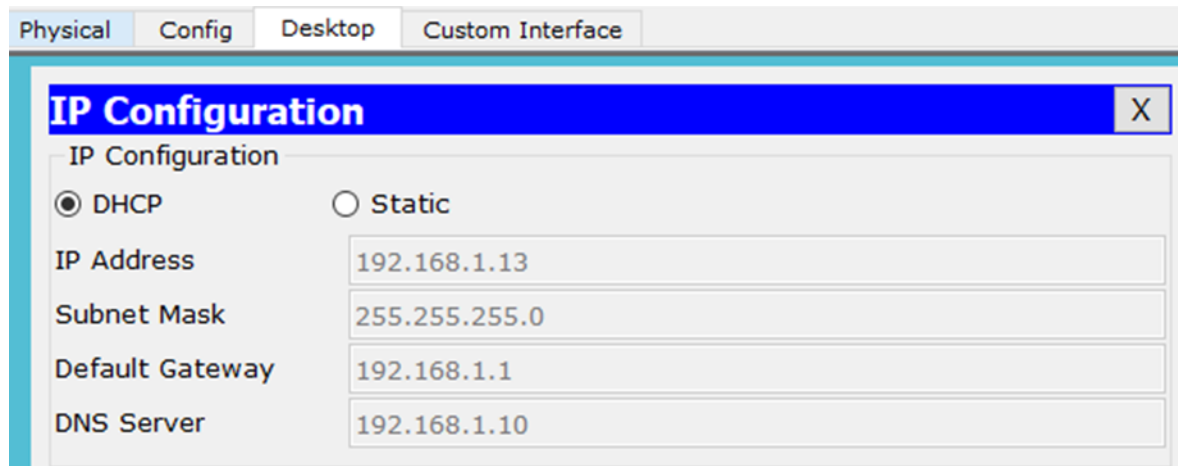
Edit the Index .html and update it



Now go to every PC and on their IP configuration tabs, enable DHCP. Every PC should be able to obtain an IP address, default gateway and DNS server, as defined in step 2.

For example, to enable DHCP on PC1:

Click PC1->Desktop->IP configuration. Then enable DHCP:



The screenshot shows a window titled "IP Configuration" with a close button (X) in the top right corner. The window has a tabbed interface with four tabs: "Physical", "Config", "Desktop", and "Custom Interface". The "Desktop" tab is selected. Inside the "Desktop" tab, there is a section titled "IP Configuration". Under this section, there are two radio buttons: "DHCP" (which is selected) and "Static". Below the radio buttons, there are four text input fields with the following values:

Field	Value
IP Address	192.168.1.13
Subnet Mask	255.255.255.0
Default Gateway	192.168.1.1
DNS Server	192.168.1.10

Do this for the other PCs.

You can test the configuration by pinging PC2 from PC1. Ping should succeed.

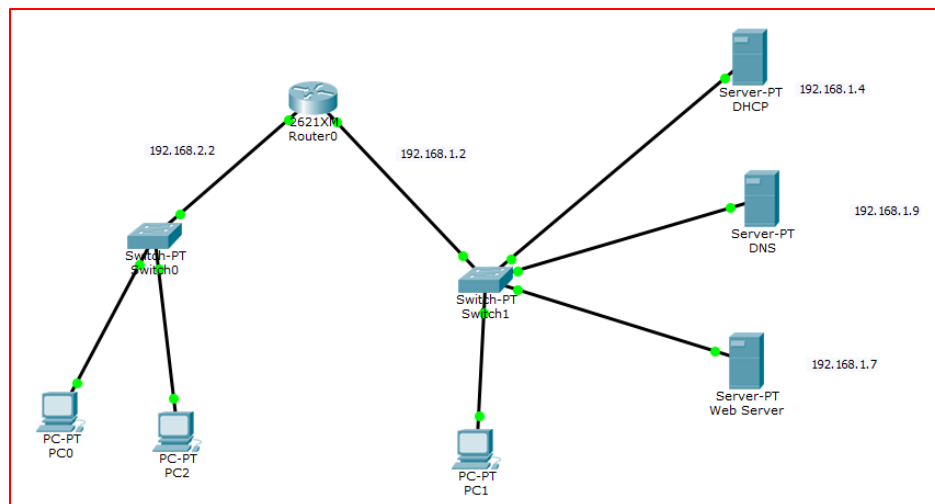
It's that simple!

Lab Tasks

Experiment 1: DNS Server Configuration in Packet Tracer

- Set up a network topology in Packet Tracer that includes a DNS server, client devices, and a local network.
- Configure the DNS server with a static IP address and assign a domain name to the server.
- Configure the client devices to use the DNS server for name resolution.
- Verify the DNS resolution of domain names to IP addresses and vice versa.

Experiment 2: Design, label and Configure the following topology in cisco packet tracer



- Set up a network topology in Packet Tracer
- Configure a generic server as centralized DHCP server to provide IP addresses to Network 192.168.1.0 /24 and 192.168.2.0/24
- Configure the client devices to use the DNS server for name resolution.
- Validate that the DNS resolution process correctly translates domain names into IP addresses for the web server.