

Networking Lab - 6

OBJECTIVE:

1. Setup DNS on a server.
2. Update DNS settings on systems.
3. Setup HTTP web-server on server.
4. Test website.

STEPS:

- Start the packet tracer file included (Lab-6 Start), and have a look at the configuration.
- Firstly, change the DNS server to the name of the server in the IP Configuration of the server.
- Go to DHCP in services, update the DNS to the same, and Starting IP address.
- Now go to each device in their command prompt, and run “ipconfig/renew”.
- Make an entry for DNS in the server, and turn ON the HTTP server.
- Verify by going to web-browser in PC#1 and hit the domain name for the DNS server.
- Alternate method is to ping the server in command prompt using ip address or the domain name.

DNS-The History

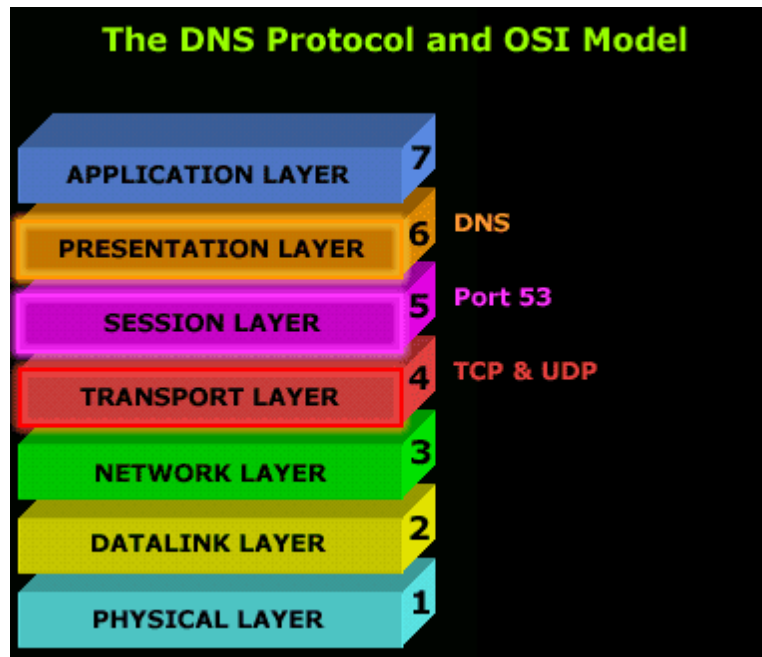
DNS began in the early days when the Internet was only a small network created by the Department of Defence for research purposes. Host names (simple computer names) of computers were manually entered into a file (called HOSTS) which was located on a central server. Each site/computer that needed to resolve host names had to download this file. But as the number of hosts grew, so did the HOSTS file (Linux, Unix, Windows and NetWare still use such files) until it was far too large for computers to download and it was generating great amounts of traffic ! So they thought ... Stuff this .. let's find a better solution ... and in 1984 the Domain Name System was introduced.

The Protocol

The **Domain Name System** is a 'hierarchically distributed database', which is a fancy way of saying that its layers are arranged in a definite order and that its data is distributed across a wide range of machines (just like the roots of a tree branch out from the main root).

Most companies today have their own little DNS server to ensure the computers can find each other without problems. If you're using Windows 2000 and Active Directory, then you surely are using DNS for the name resolutions of your computers. Microsoft has created its own version of a "DNS" server, called a WINS server, which stands for Windows Internet Name Service, but this is old technology and uses protocols that are nowhere near as efficient as DNS, so it was natural for Microsoft to move away from WINS and towards DNS, after all, the whole Internet works on DNS :)

The DNS protocol works when your computer sends out a DNS query to a name server to resolve a domain. For example, you type "www.firewall.cx" in your web browser, this triggers a DNS request, which your computer sends to a DNS server in order to get the website's IP Address ! There is a detailed example on the pages to follow so I won't get into too much detail for the moment.



The DNS protocol normally uses the UDP protocol as a means of transport because of its small overhead in comparison to TCP; the less overhead a protocol has, the faster it is !

In the case where there are constant errors and the computer trying to request a DNS resolution can't get an error free answer, or any answer at all, it will switch to TCP to ensure the data arrives without errors.

This process, though, depends on the operating system you're using. Some operating systems might not allow DNS to use the TCP protocol, thus limiting it to UDP only. It is rare that you will get so many errors that you can't resolve any hostname or domain name to an IP Address.

The **DNS protocol** utilises **Port 53** for its service. This means that a DNS server listens on Port 53 and expects any client wishing to use the service to use the same port. There are, however, cases where you might need to use a different port, something possible depending on the operating system and DNS server you are running.

In the following pages we'll be looking at the actual DNS packet format, where you are able to see exactly the contents of DNS query, so we won't analyse the packet structure here.

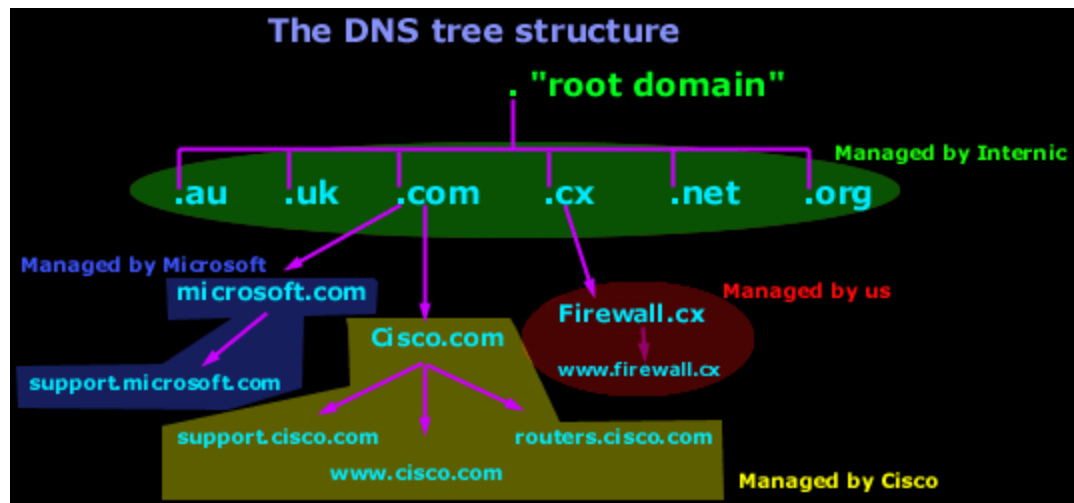
Next we'll take a close look at how the Internet domains and DNS servers are structured to make sure the model works flawlessly and efficiently!

The Internet Domain Name Server Hierarchy

This interesting section will help you understand how domain names on the Internet are structured and where DNS servers fit in to the picture. When you think about the millions of domain names registered today, you probably think that you have to be superhuman to manage such a structure of DNS servers !

Well that's not that case. The DNS structure has been designed in such a way that no DNS server needs to know about all possible domains, but only those immediately above and below it.

The picture below shows part of the internet **DNS hierarchical structure**:



Let's explain how it works:

Internic controls the "root" domain, which includes all the top level domains. These are marked in a green oval for clarity. Within the green oval you have the ROOT DNS servers, which know all about the authoritative DNS servers for the domains immediately below them e.g firewall.cx, cisco.com, microsoft.com etc. These ROOT DNS servers can tell you which DNS server takes care of firewall.cx, cisco.com, microsoft.com and the rest.

Each domain, including the ones we are talking about (cisco, firewall, microsoft), have what we call a "Primary DNS" and "Secondary DNS". The Primary DNS is the one that holds all the information about its domain. The Secondary acts as a backup in case the Primary DNS fails. The process in which a Primary DNS server sends its copy to the Secondary DNS server is called *Zone Transfer* and is covered in the DNS Database section.

Today there are hundreds of websites at which you are able to register your own domain and, once you've done that, you have the power to manage it yourself. In the example above, Cisco bought the "Cisco.com"

domain and then created your resource records. Some examples of resource records for the Cisco domain in our example are: support , www and routers. These will be analysed in depth on the next pages.

So here comes the million dollar question: **How do you create subdomains and www's (known as resource records) ?**

The answer is pretty simple:

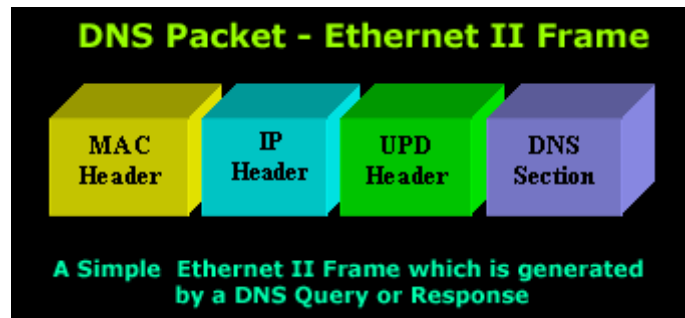
You use a special DNS administration interface (usually web based - provided by the guys with whom you registered your domain) that allows you to create, change and delete the subdomains, www's or whatever resource record you can come up with. When you're making changes to the DNS settings of your domain, you're actually changing the contents of specific files that are located on that server.

These changes then slowly propagate to the *authoritative* DNS servers, which are responsible for your domain area and then the whole Internet will contact these DNS servers when they need to access any section of your domain.

For example, if you need to resolve ftp.firewall.cx, your computer will locate and contact the DNS Server responsible for the .CX domains, which will let you know the DNS server that's in charge of the Firewall.cx domain. The DNS server of Firewall.cx in turn will let your computer know the IP Address of ftp.firewall.cx because it holds all the information for the firewall.cx domain

Analyzing a DNS Packet

Following is the Ethernet II packet that runs on the local network. The structure is the same as our previous DNS query packet, but varies in size:



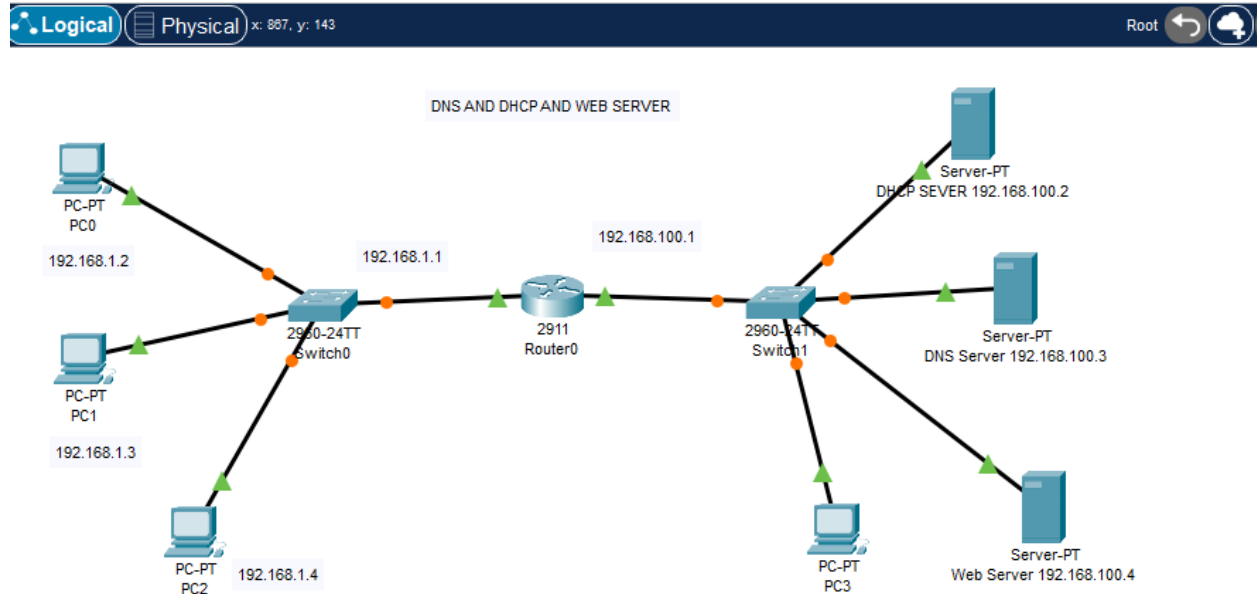
Now, to make the analysis of the DNS Section easier we have also included the **DNS Query** (left) and **DNS Response** (right). This allows us to easily compare both **DNS query and response packets**:

HOW CONFIGURE DNS SERVER STEP BY STEP:-

Domain Name System (**DNS**) server resolves host names into IP addresses. Although we can access a network host using its IP address, DNS makes it easier by allowing us use domain names which are easier to remember. For example its much easier to access google website by typing `http://www.google.com` as compared to

Now, before any host can use a DNS service, we must configure a DNS server first. For example, when you type the URL `http://www.google.com` in your browser, the host will query the DNS server for the IP address of `http://www.google.com`. The DNS server will resolve `http://www.google.com` into an IP address then answer back the host with the IP address.

1.Build the network topology.

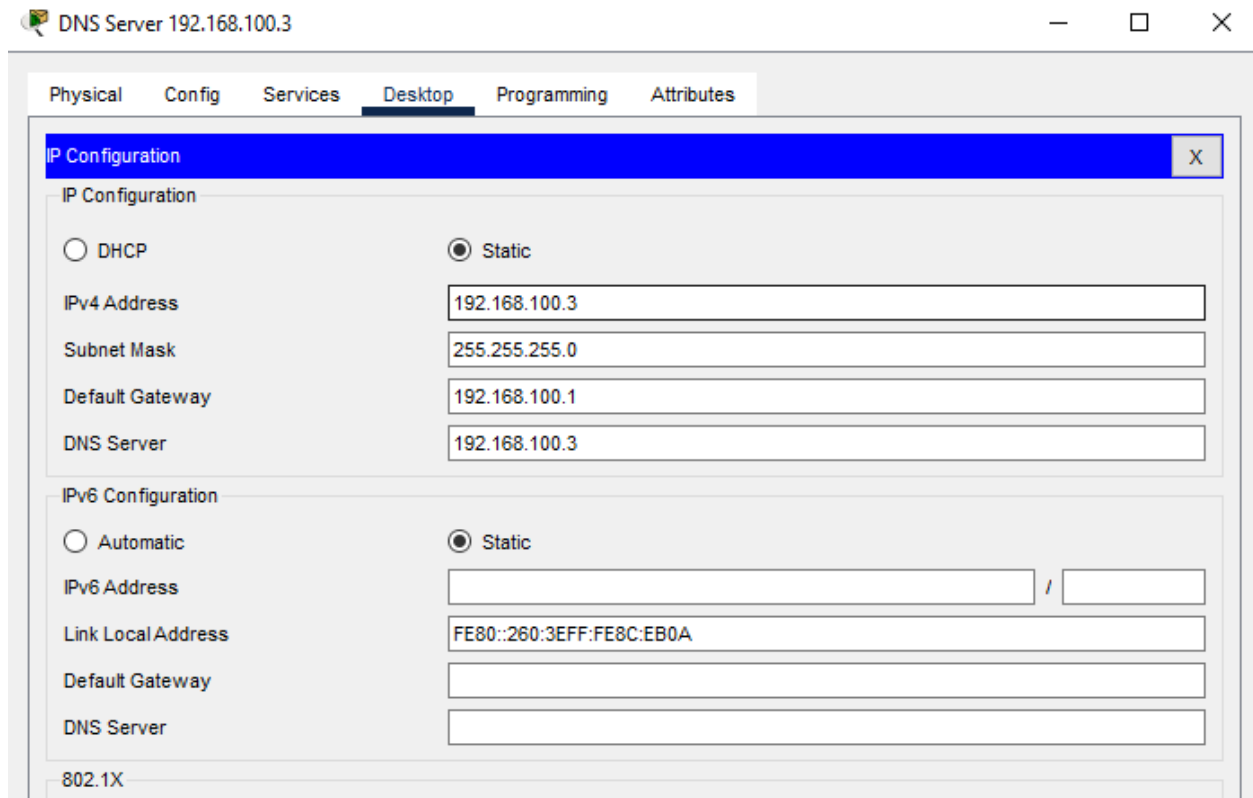


Step 1:-

Firstly, change the DNS server to the name of the server in the IP Configuration of the server.

Configure static IP addresses on the PCs and the server.

DNS Server



The screenshot shows a window titled "DNS Server 192.168.100.3" with a standard Windows interface (minimize, maximize, close buttons). The window has a tabbed interface with the following tabs: Physical, Config, Services, Desktop (selected), Programming, and Attributes. The "Desktop" tab is active, and within it, the "IP Configuration" section is highlighted in blue. This section contains two sub-sections: "IP Configuration" and "IPv6 Configuration".

IP Configuration

- ☐ DHCP
- ☒ Static
- IPv4 Address: 192.168.100.3
- Subnet Mask: 255.255.255.0
- Default Gateway: 192.168.100.1
- DNS Server: 192.168.100.3

IPv6 Configuration

- ☐ Automatic
- ☒ Static
- IPv6 Address: [Empty field] / [Empty field]
- Link Local Address: FE80::260:3EFF:FE8C:EB0A
- Default Gateway: [Empty field]
- DNS Server: [Empty field]

At the bottom of the window, the text "802.1X" is visible.

Physical **Config** Services Desktop Programming Attributes**GLOBAL**

Settings

Algorithm Settings

INTERFACE

FastEthernet0

Global Settings

Display Name

Gateway/DNS IPv4

☐ DHCP☒ StaticDefault Gateway DNS Server

Gateway/DNS IPv6

☐ Automatic☒ StaticDefault Gateway DNS Server

DNS SERVER CONFIGURE A NAME AND IP ADDRESS AND DNS SERVICE ON

DNS Server 192.168.100.3

Physical Config **Services** Desktop Programming Attributes

SERVICES ^

- HTTP
- DHCP
- DHCPv6
- TFTP
- DNS**
- SYSLOG
- AAA
- NTP
- EMAIL
- FTP
- IoT
- VM Management
- Radius EAP

DNS

DNS Service ☒ On ☐ Off

Resource Records

Name Type **A Record** v

Address

Add **Save** **Remove**

No.	Name	Type	Detail
0	google.com	A Record	192.168.100.4

DHCP SERVER:-

Why use DHCP?

Every device on a TCP/IP-based network must have a unique unicast IP address to access the network and its resources. Without DHCP, IP addresses for new computers or computers that are moved from one subnet to another must be configured manually; IP addresses for computers that are removed from the network must be manually reclaimed.

With DHCP, this entire process is automated and managed centrally. The DHCP server maintains a pool of IP addresses and leases an address to any DHCP-enabled client when it starts up on the network. Because the IP addresses are dynamic (leased) rather than static (permanently assigned), addresses no longer in use are automatically returned to the pool for reallocation.

The network administrator establishes DHCP servers that maintain TCP/IP configuration information and provide address configuration to DHCP-enabled clients in the form of a lease offer. The DHCP server stores the configuration information in a database that includes:

- Valid TCP/IP configuration parameters for all clients on the network.
- Valid IP addresses, maintained in a pool for assignment to clients, as well as excluded addresses.
- Reserved IP addresses associated with particular DHCP clients. This allows consistent assignment of a single IP address to a single DHCP client.
- The lease duration, or the length of time for which the IP address can be used before a lease renewal is required.

A DHCP-enabled client, upon accepting a lease offer, receives:

- A valid IP address for the subnet to which it is connecting.
- Requested DHCP options, which are additional parameters that a DHCP server is configured to assign to clients. Some examples of DHCP options are Router (default gateway), DNS Servers, and DNS Domain Name.

Benefits of DHCP

DHCP provides the following benefits.

- **Reliable IP address configuration.** DHCP minimizes configuration errors caused by manual IP address configuration, such as typographical errors, or address conflicts caused by the assignment of an IP address to more than one computer at the same time.
- **Reduced network administration.** DHCP includes the following features to reduce network administration:
 - Centralized and automated TCP/IP configuration.
 - The ability to define TCP/IP configurations from a central location.
 - The ability to assign a full range of additional TCP/IP configuration values by means of DHCP options.
 - The efficient handling of IP address changes for clients that must be updated frequently, such as those for portable devices that move to different locations on a wireless network.
 - The forwarding of initial DHCP messages by using a DHCP relay agent, which eliminates the need for a DHCP server on every subnet.

2ND STEPS DHCP SERVER CONFIGURE

DHCP SEVER 192.168.100.2

Physical Config Services **Desktop** Programming Attributes

IP Configuration X

IP Configuration

☐ DHCP ☒ Static

IPv4 Address 192.168.100.2

Subnet Mask 255.255.255.0

Default Gateway 192.168.100.1

DNS Server 192.168.100.3

IPv6 Configuration

☐ Automatic ☒ Static

IPv6 Address /

Link Local Address FE80::260:2FFF:FE10:5D37

Default Gateway

DNS Server

802.1X

☐ Use 802.1X Security

Authentication MD5

Username

Password

Physical **Config** Services Desktop Programming Attributes**GLOBAL**

Settings

Algorithm Settings

INTERFACE

FastEthernet0

Global Settings

Display Name DHCP SEVER 192.168.100.2

Gateway/DNS IPv4

☐ DHCP☒ Static

Default Gateway 192.168.100.1

DNS Server 192.168.100.3

Gateway/DNS IPv6

☐ Automatic☒ Static

Default Gateway

DNS Server

DHCP SERVICE CONFIGURE

DHCP SERVER 192.168.100.2

Physical Config **Services** Desktop Programming Attributes

SERVICES

- HTTP
- DHCP**
- DHCPv6
- TFTP
- DNS
- SYSLOG
- AAA
- NTP
- EMAIL
- FTP
- IoT
- VM Management
- Radius EAP

DHCP

Interface: FastEthernet0 Service: ☒ On ☐ Off

Pool Name: serverPool

Default Gateway: 0.0.0.0

DNS Server: 0.0.0.0

Start IP Address: 192 168 100 0

Subnet Mask: 255 255 255 0

Maximum Number of Users: 512

TFTP Server: 0.0.0.0

WLC Address: 0.0.0.0

Add Save Remove

Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server	WLC Address
serverPool1	192.168....	192.168....	192.168....	255.255....	50	0.0.0.0	0.0.0.0
serverPool2	192.168....	192.168....	192.168....	255.255....	50	0.0.0.0	0.0.0.0
serverPool	0.0.0.0	0.0.0.0	192.168....	255.255....	512	0.0.0.0	0.0.0.0

ServerPool1 –Default Gateway- 192.168.100.1 ,DNS Server :- 192.168.100.3 , Subnet Mask- 255.255.255.0 , Maximum User – 50.

Web Server:-

A web server is a software application or hardware device that stores, processes, and serves web content to users over the internet. It plays a critical role in the client-server model of the World Wide Web, where clients (typically web browsers) request web pages and resources, and servers respond to these requests by delivering the requested content.

Web servers operate on the Hypertext Transfer Protocol (HTTP), which is the foundation of data communication on the World Wide Web. When you enter a website's URL into your browser, it sends an HTTP request to the web server hosting that website, which then sends back the web page you requested, allowing you to view it in your browser.

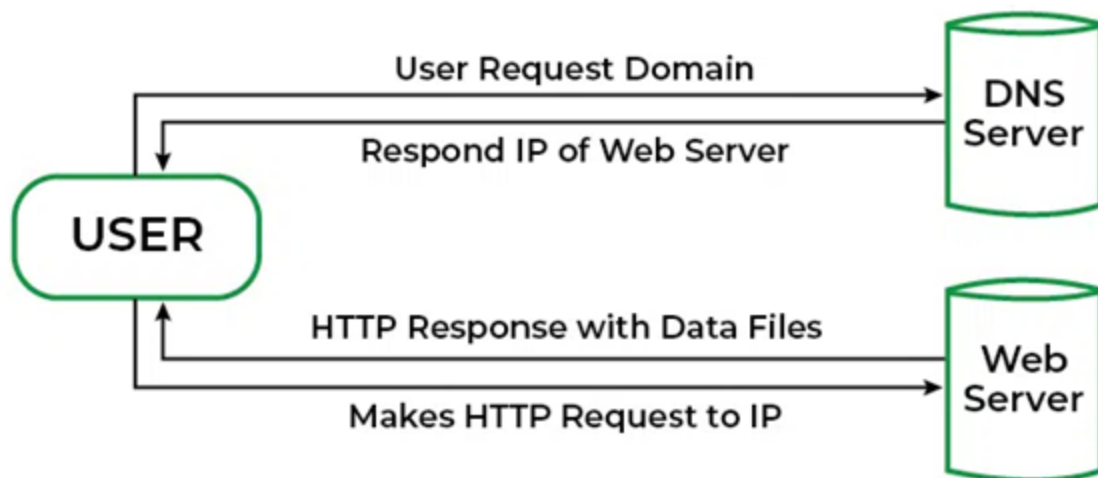
Web Server Architecture

Web server architecture refers to the structure and design of web servers, outlining how they handle incoming requests and deliver web content. There are two main approaches to web server architecture:

- **Single-Tier (Single Server) Architecture:**

In a single-tier architecture, a single server is responsible for both processing requests and serving web content. This is suitable for small websites or applications with low traffic. However, it has limitations in terms of scalability and fault tolerance. If the server goes down, the entire service becomes unavailable.

Single Server Architecture of Web Server



How to Create Web Server

Step1:- Configure IP Address-

Web Server 192.168.100.4

Physical Config Services **Desktop** Programming Attributes

IP Configuration X

IP Configuration

☐ DHCP ☒ Static

IPv4 Address

Subnet Mask

Default Gateway

DNS Server

IPv6 Configuration

☐ Automatic ☒ Static

IPv6 Address /

Link Local Address

Default Gateway

DNS Server

802.1X

☐ Use 802.1X Security

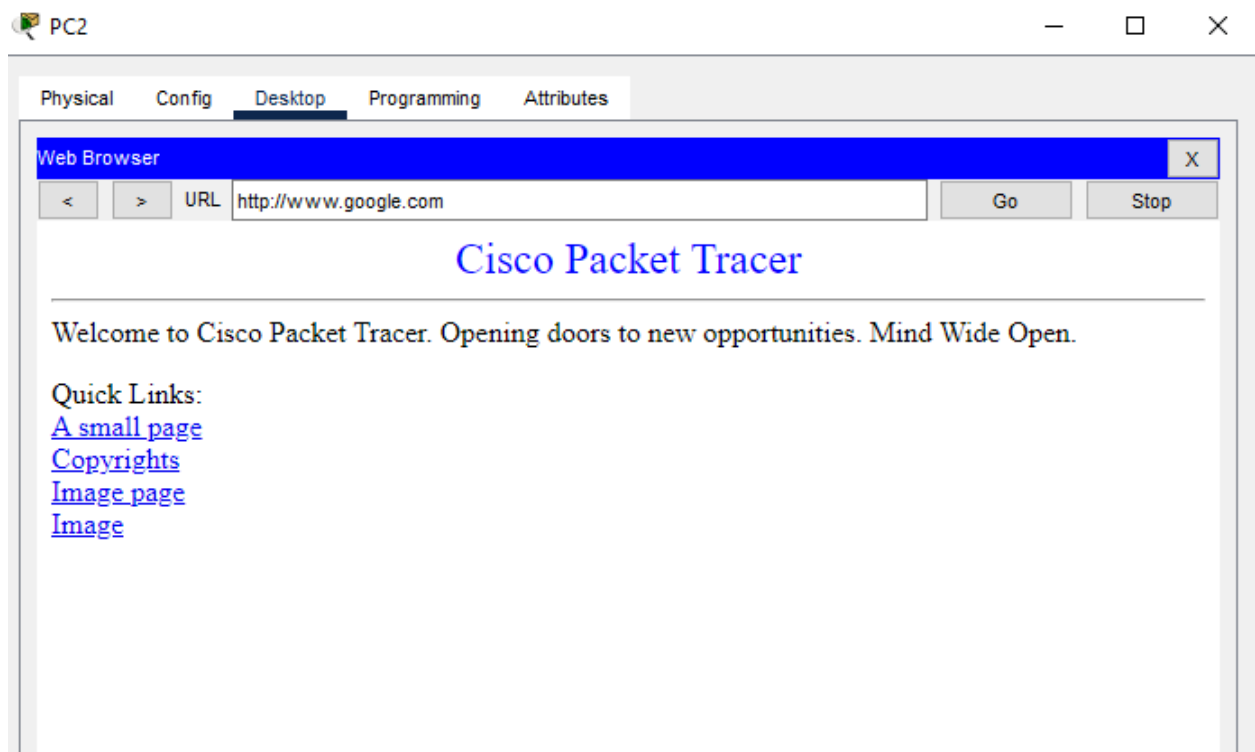
Authentication

Username

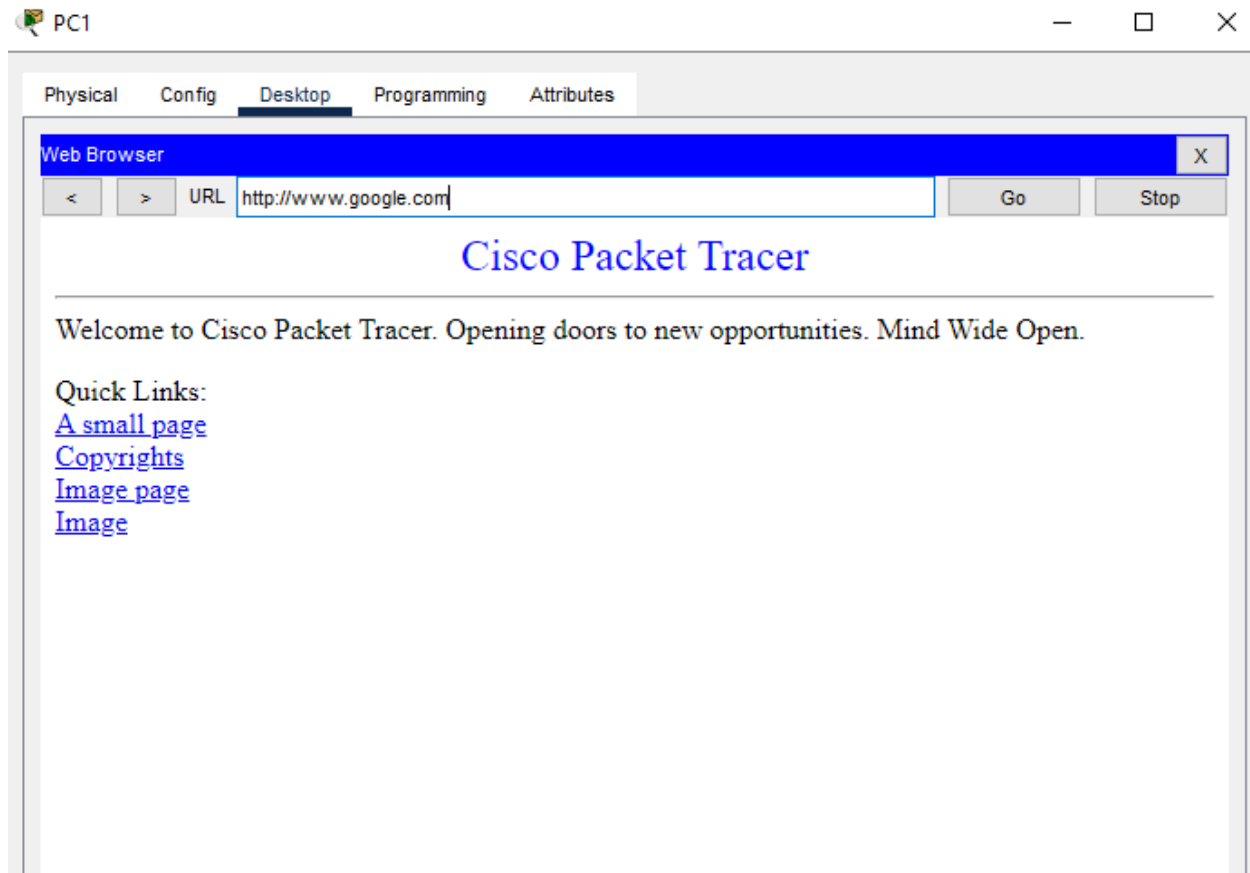
Password

Pc2 – Verify by going to web-browser in PC#2 and hit the domain name for the DNS server.

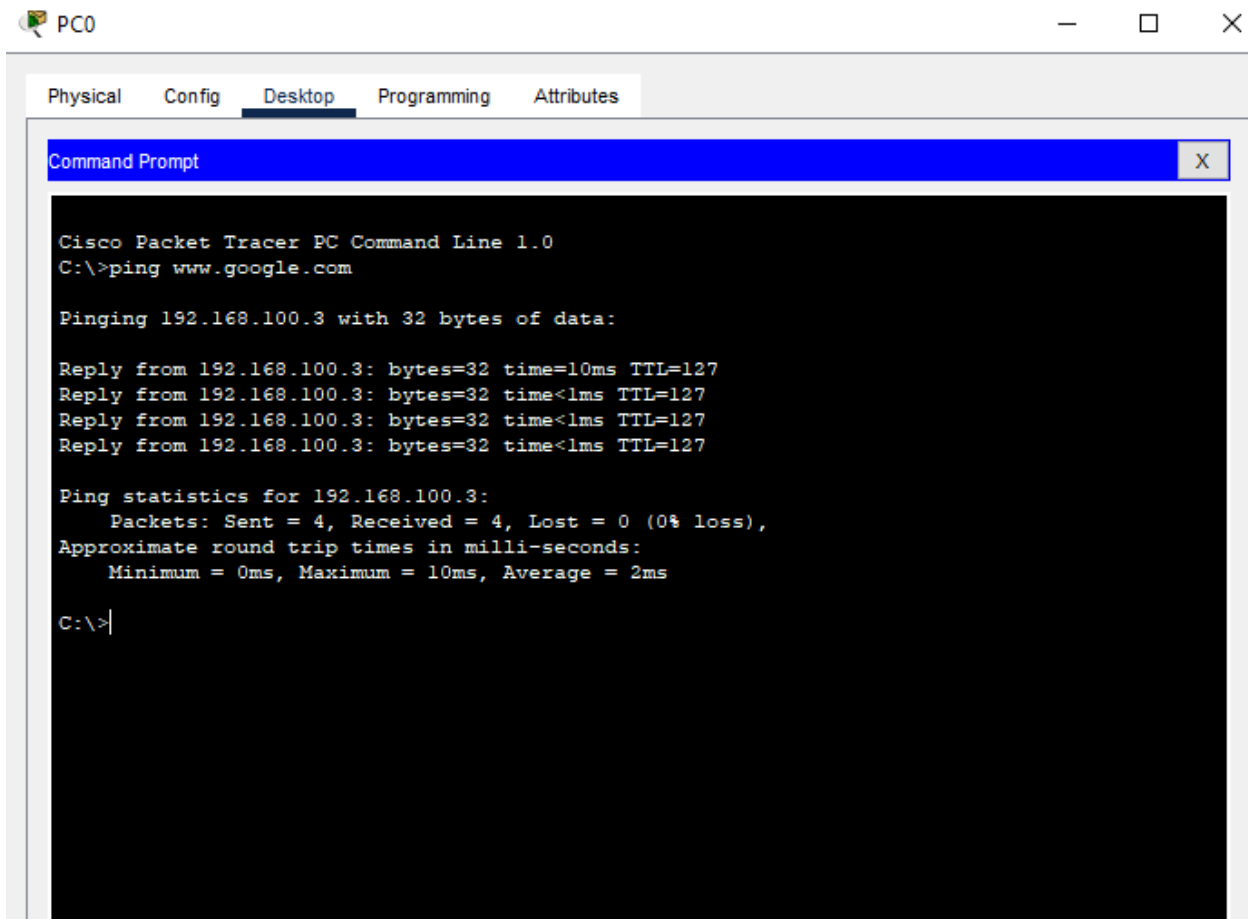
Web Test- www.google.com



Pc-1 – Verify by going to web-browser in PC#1 and hit the domain name for the DNS server.



Alternate method is to ping the server in command prompt using ip address or the domain name



The screenshot shows a Cisco Packet Tracer interface with a PC named 'PC0'. The 'Desktop' tab is selected, displaying a 'Command Prompt' window. The command prompt shows the execution of the command 'ping www.google.com', which results in four successful replies from IP address 192.168.100.3. The ping statistics indicate 0% loss and an average round trip time of 2ms.

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping www.google.com

Pinging 192.168.100.3 with 32 bytes of data:

Reply from 192.168.100.3: bytes=32 time=10ms TTL=127
Reply from 192.168.100.3: bytes=32 time<1ms TTL=127
Reply from 192.168.100.3: bytes=32 time<1ms TTL=127
Reply from 192.168.100.3: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.100.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 10ms, Average = 2ms

C:\>|
```