Computer Networks Lab
Spring 2024
Week 03 and 04

HTTP, DNS, and DHCP services

What is a Packet?

A "packet" is a single message from any network protocol (e.g., TCP, DNS, etc.).

What mode is LAN traffic in?

LAN traffic is in broadcast mode, meaning a single computer with Wireshark can see traffic between two other computers. To see traffic to an external site, you need to capture the packets on the local computer.

What is Wireshark?

Wireshark is an open-source network protocol analysis software program, widely considered the industry standard. A global organization of network specialists and software developers supports Wireshark and continues to make updates for new network technologies and encryption methods. Government agencies, corporations, non-profits, and educational institutions use Wireshark for troubleshooting and teaching purposes. There truly isn't a better way to learn low-level networking than to look at traffic under the Wireshark microscope. You should only use Wireshark on networks where you have permission to inspect network packets. Using Wireshark to look at packets without permission is illegal.

How does Wireshark work?

Wireshark is a packet sniffer and analysis tool. It captures network traffic from ethernet, Bluetooth, wireless (IEEE.802.11), token ring, and frame relay connections, among others, and stores that data for offline analysis. Wireshark allows you to filter the log before the capture starts or during analysis, so you can narrow down and zero in on what you're looking for in the network trace. For example, you can set a filter to see TCP traffic between two IP addresses, or you can set it only to show you the packets sent from one computer. The filters in Wireshark are one of the primary reasons it has become the standard tool for packet analysis.

When should Wireshark be used?

Wireshark can be used to understand how communication takes place across a network and to analyze what went wrong when an issue in communication arises.

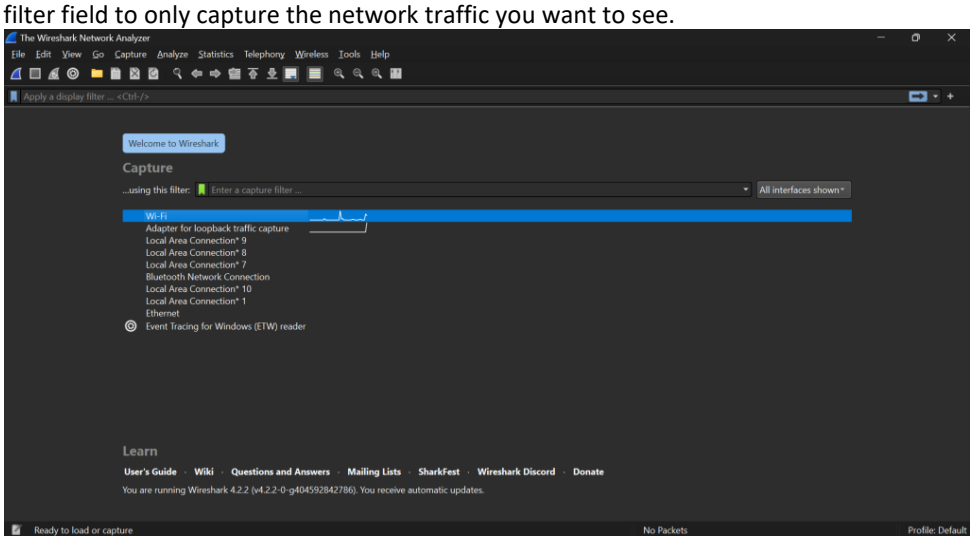
Wireshark helps:

- Network administrators troubleshoot problems across a network
- Security engineers examine security issues across a network
- QA engineers verify applications
- Developers debug protocol implementations
- Network users learn about a specific protocol

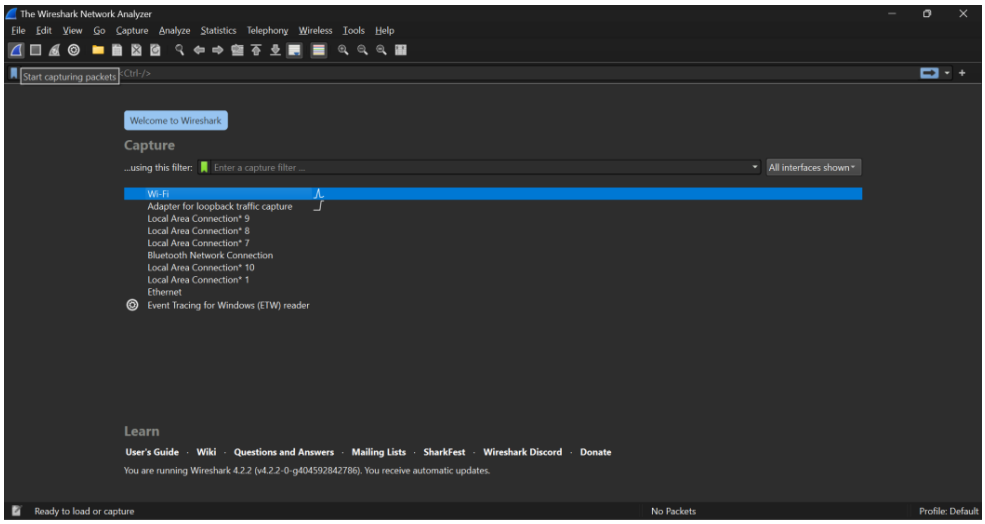
Data packets on Wireshark

Capturing data packets on Wireshark

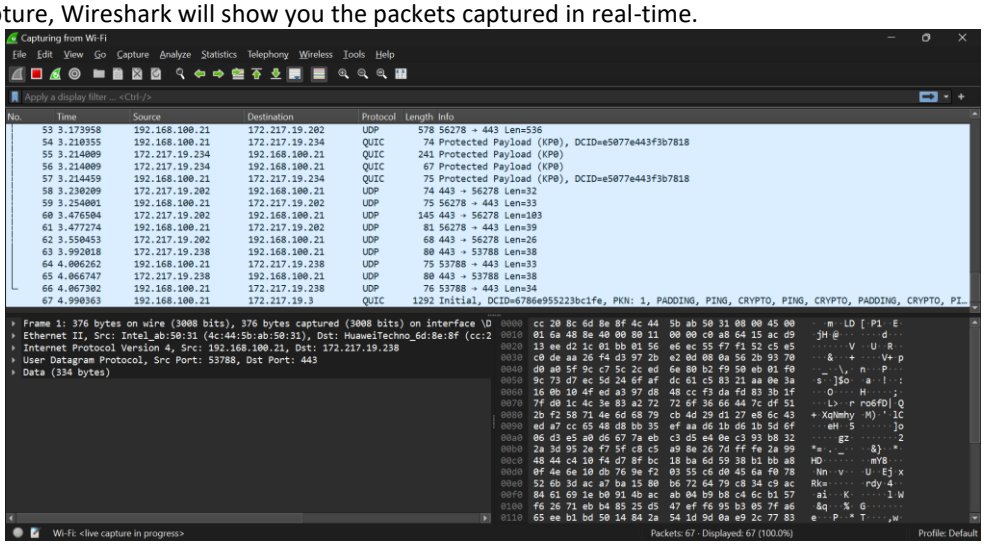
When you open Wireshark, you see a screen showing you a list of all the network connections you can monitor. You also have a capture filter field to only capture the network traffic you want to see.



You can select one or more of the network interfaces using shift+left-click. Once select the network interface, you can start the capture, and there are several ways to do that. Click the first button on the toolbar, titled “Start capturing packets.”



During the capture, Wireshark will show you the packets captured in real-time.



Once you have captured all the packets needed, use the red button (next to the start one) or menu options to stop the capture as you did to begin. Best practice dictates stopping Wireshark's packet capture before analysis.

Analyzing data packets on Wireshark

Wireshark shows you three different panes for inspecting packet data. The Packet List, the top pane, lists all the packets in the capture. When you click on a packet, the other two panes change to show you the details about the selected packet. You can also tell if the packet is part of a conversation. Here are details about each column in the top pane:

- No.: This is the number order of the packet captured. The bracket indicates that this packet is part of a conversation.
- Time: This column shows how long after you started the capture this particular packet was captured. You can change this value in the Settings menu to display a different option.
- Source: This is the address of the system that sent the packet.
- Destination: This is the address of the packet destination.
- Protocol: This is the type of packet. For example: TCP, DNS, DHCPv6, or ARP.
- Length: This column shows you the packet's length, measured in bytes.
- Info: This column shows you more information about the packet contents, which will vary depending on the type of packet.
- Packet Details, the middle pane, shows you as much readable information about the packet as possible, depending on the packet type. You can right-click and create filters based on the highlighted text in this field.
- The bottom pane, Packet Bytes, displays the packet exactly as it was captured in hexadecimal.

When looking at a packet that is part of a conversation, you can right-click the packet and select Follow to see only the packets that are part of that conversation.

Wireshark capture filters

Capture filters limit the captured packets by the chosen filter. If the packets don't match the filter, Wireshark won't save them. Examples of capture filters include:

- host IP-address: This filter limits the captured traffic to and from the IP address
- net 192.168.0.0/24: This filter captures all traffic on the subnet
- dst host IP-address: Capture packets sent to the specified host
- port 53: Capture traffic on port 53 only
- port not 53 and not arp: Capture all traffic except DNS and ARP traffic

Wireshark display filters

Wireshark display filters change the view of the capture during analysis. After you've stopped the packet capture, use display filters to narrow down the packets in the Packet List to troubleshoot your issue. One of the most useful display filters is:

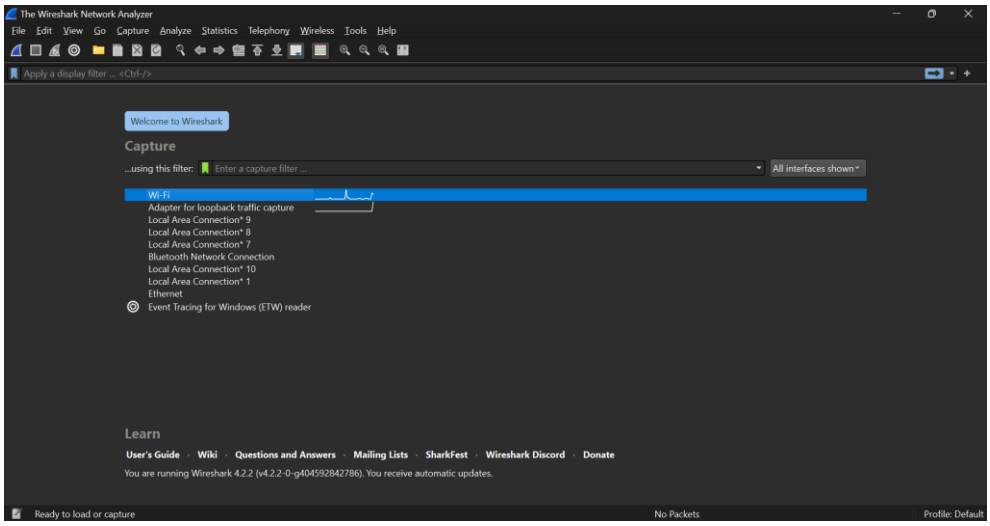
- ip.src==IP-address and ip.dst==IP-address: This filter shows packets sent from one computer (ip.src) to another (ip.dst). You can also use ip.addr to show packets to and from that IP. Other filters include:
- tcp.port eq 25: This filter will show you all traffic on port 25, which is usually SMTP traffic
- icmp: This filter will show you only ICMP traffic in the capture, most likely they are pings
- ip.addr != IP_address: This filter shows you all traffic except the traffic to or from the specified computer

Analysts even build filters to detect specific attacks, like this filter used to detect the Sasser worm:

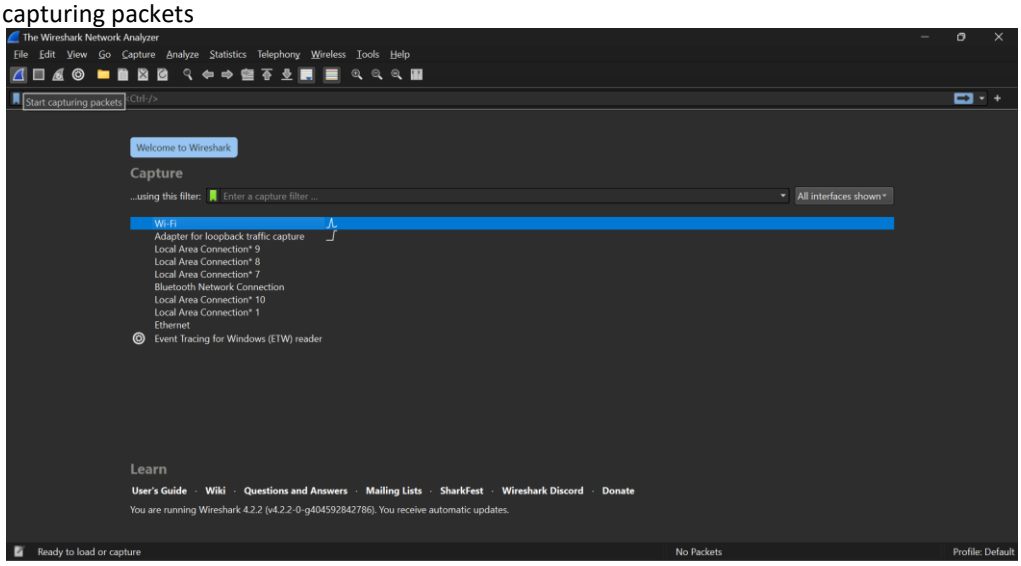
- ls_ads.opnum==0x09

Wireshark Walk through task:

Step 1: Choose the network

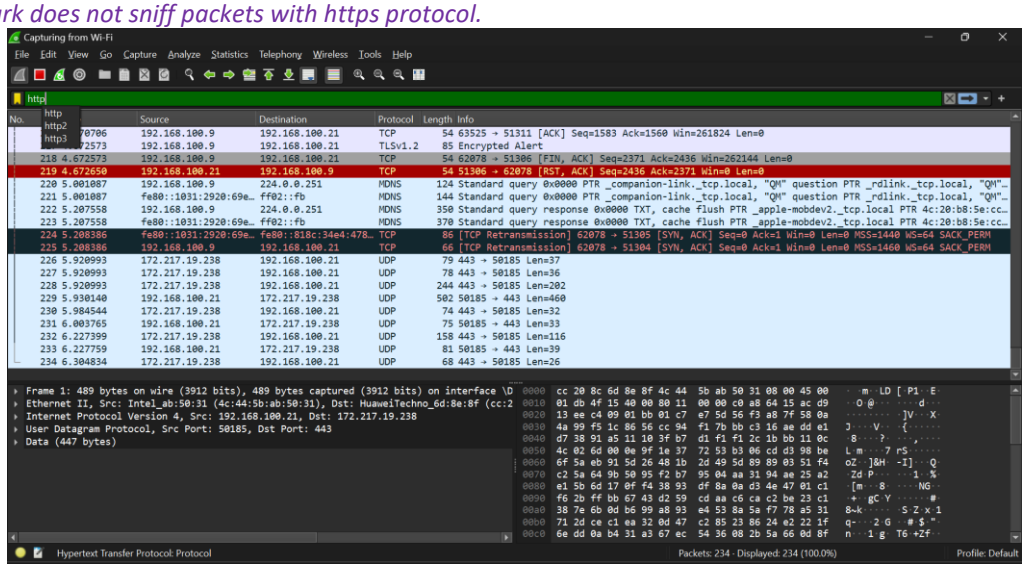


Step 2: Start capturing packets

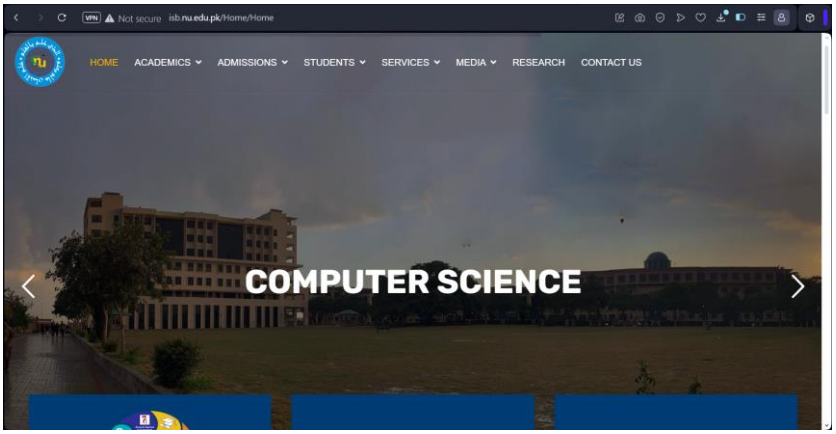


Step 3: Add a filter (http in our case), and press enter.

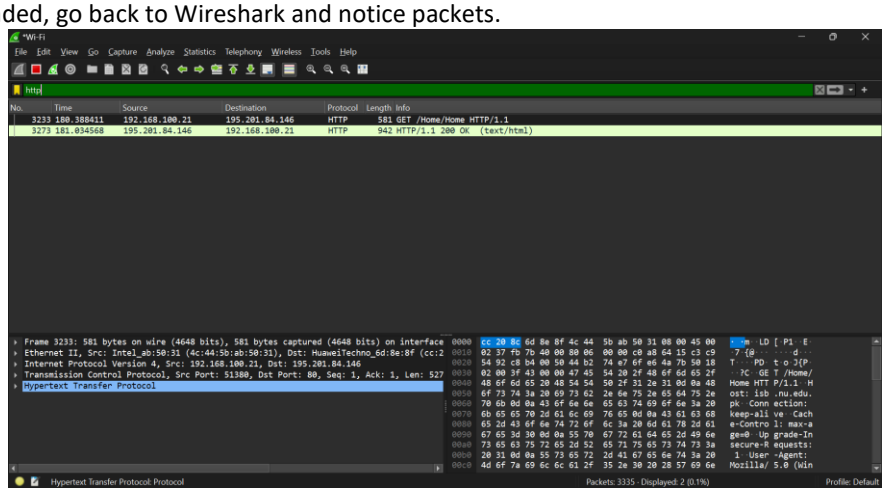
Note: wirshark does not sniff packets with https protocol.



Step 4: Go to the web browser and search for a website like:



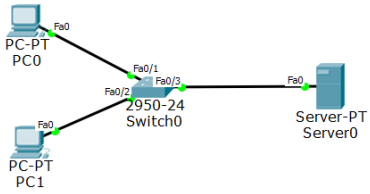
Step 5: Once all loaded, go back to Wireshark and notice packets.



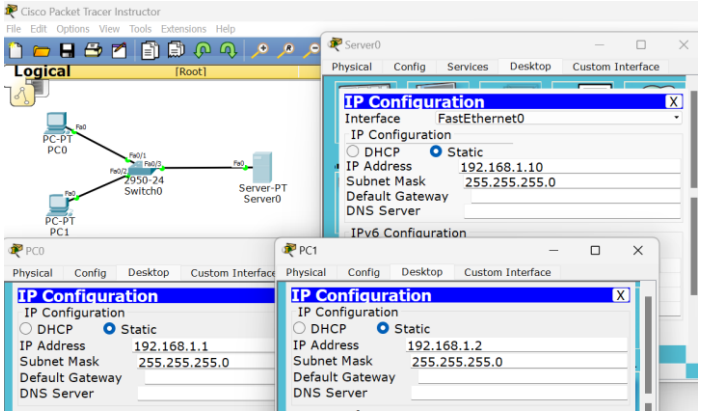
Step 6: Observe the fields

Packet Tracer Walk Through Task for HTTP:

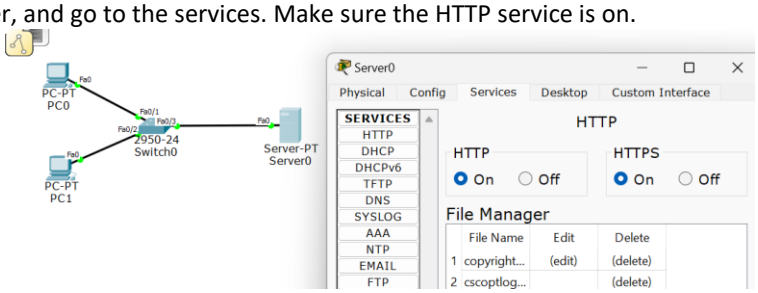
Step 1: make a topology with 2 PCs, a switch, and a server as shown



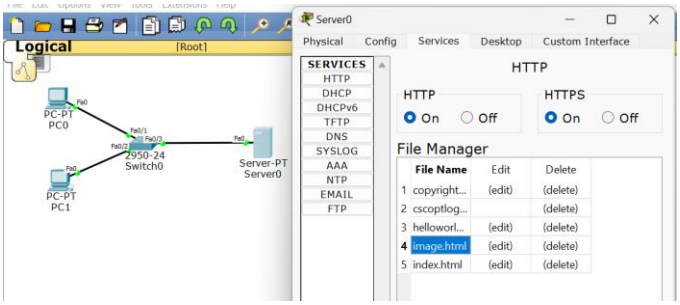
Step 2: assign ip addresses to the systems, and the server



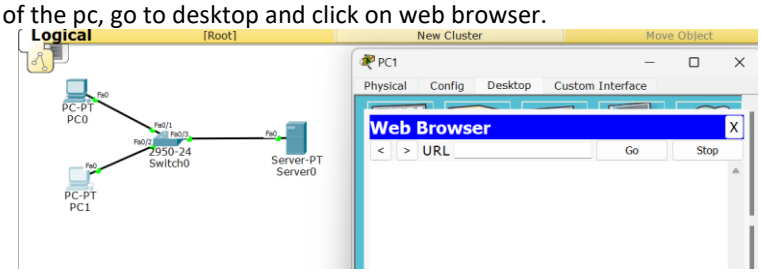
Step 3: Click on the server, and go to the services. Make sure the HTTP service is on.



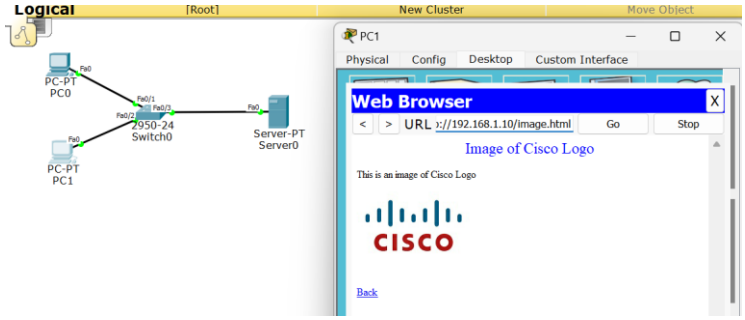
Step 4: Choose a file already available on the server



Step 5: Click on any one of the pc, go to desktop and click on web browser.



Step 6: Type in the ip address of the server, back slash, and the name of the chosen file
Like: 192.168.1.10/image.html



Dynamic Host Configuration Protocol (DHCP)

Dynamic Host Configuration Protocol (DHCP) is a client/server protocol that automatically provides an Internet Protocol (IP) host with its IP address and other related configuration information such as the subnet mask and default gateway. RFCs 2131 and 2132 define DHCP as an Internet Engineering Task Force (IETF) standard based on Bootstrap Protocol (BOOTP), a protocol with which DHCP shares many implementation details. DHCP allows hosts to obtain required TCP/IP configuration information from a DHCP server.

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.

National University of Computer & Emerging Sciences (NUCES), Islamabad

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:

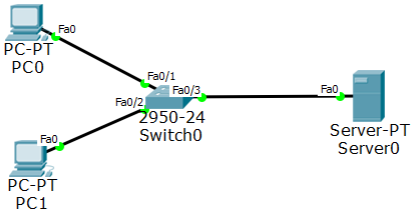
- Valid TCP/IP configuration parameters for all clients on the network.
- Valid IP addresses, maintained in a pool for assignment to clients, as well as excluded addresses.
- Reserved IP addresses associated with particular DHCP clients. This allows consistent assignment of a single IP address to a single 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:

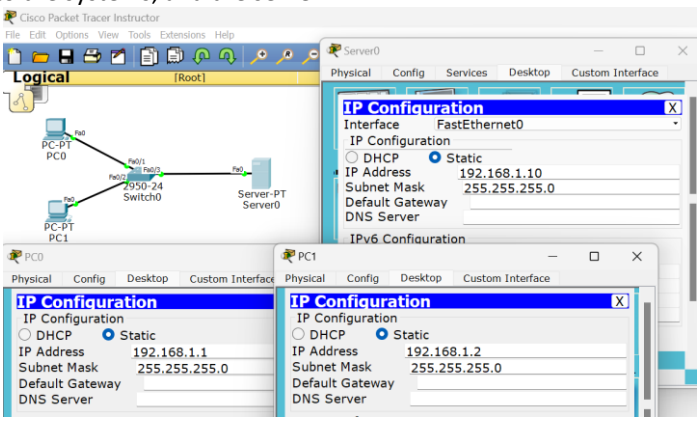
- A valid IP address for the subnet to which it is connecting.
- Requested DHCP options, which are additional parameters that a DHCP server is configured to assign to clients. Some examples of DHCP options are Router (default gateway), DNS Servers, and DNS Domain Name.

Packet Tracer Walk through task for DHCP

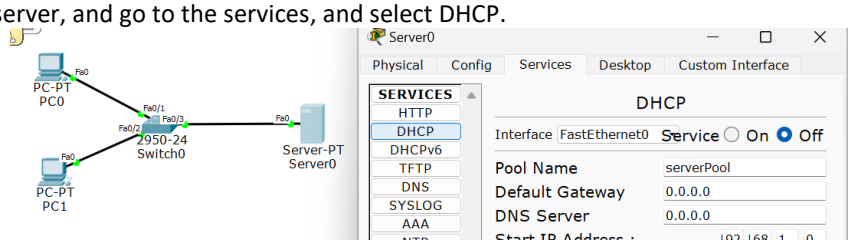
Step 1: make a topology with 2 PCs, a switch, and a server as shown



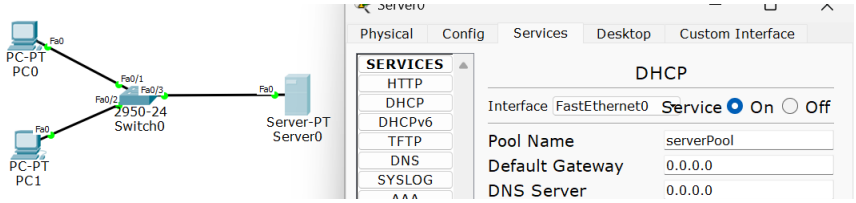
Step 2: assign ip addresses to the systems, and the server



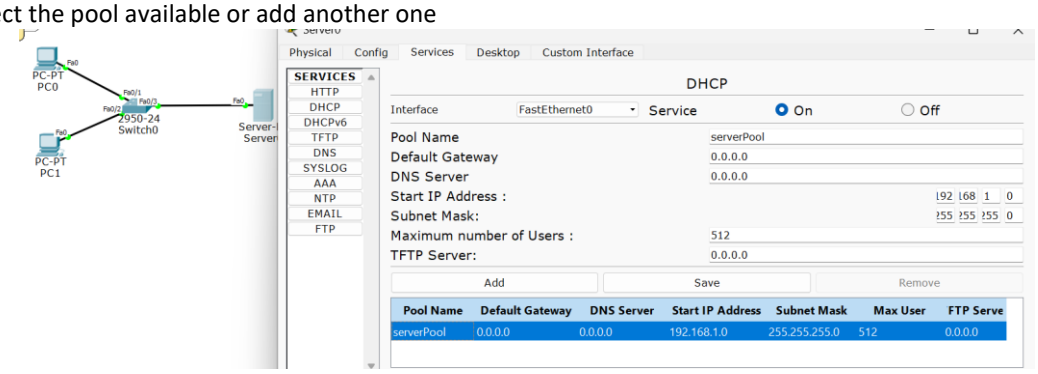
Step 3: Click on the server, and go to the services, and select DHCP.



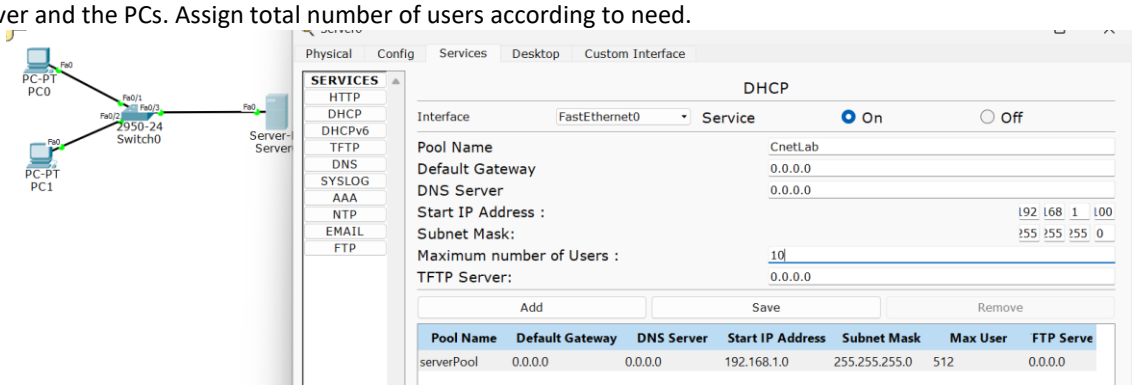
Step 4: Turn on the service



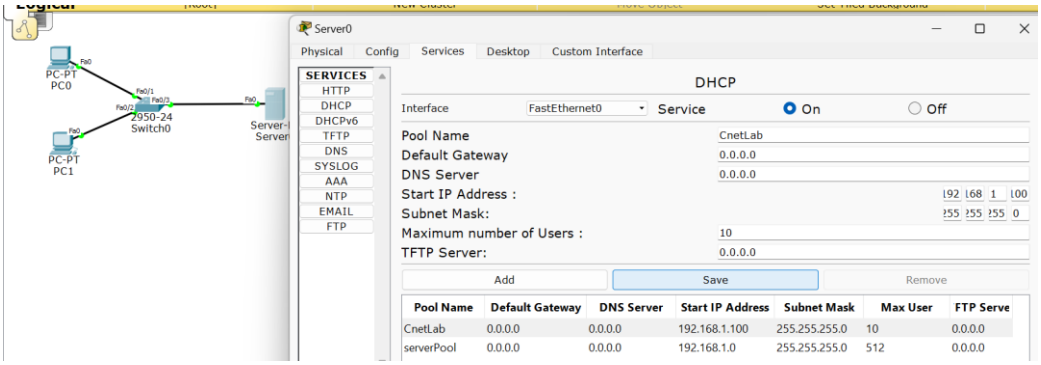
Step 5: select the pool available or add another one



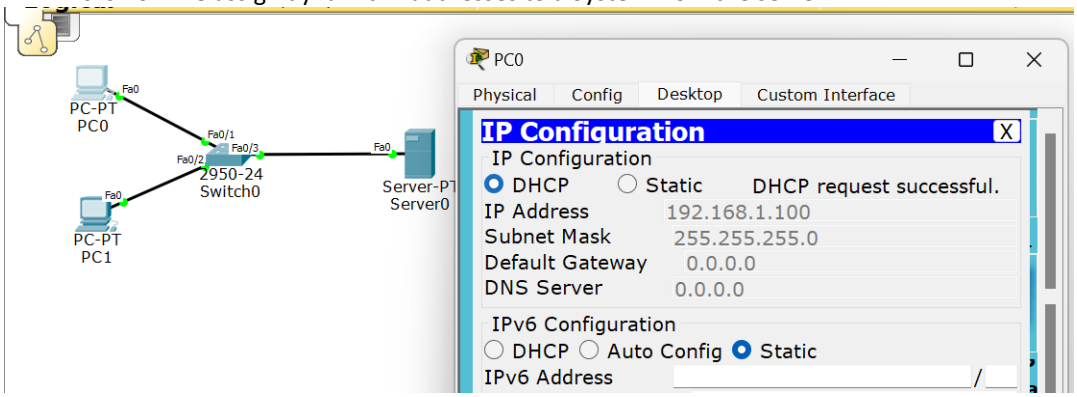
Step 6: Change the pool name, assign a starting IP address making sure it is from the same network as the ip address of the server and the PCs. Assign total number of users according to need.



Step 7: Click save



Step 8: Go to the first system, and then go to the desktop portion, and then the IP configuration. Click on the DHCP option rather than the Static one. After a few moments the server will assign a IP address from the pool and a successful message will be shown. This is how we assign dynamic IP addresses to a System from the server.

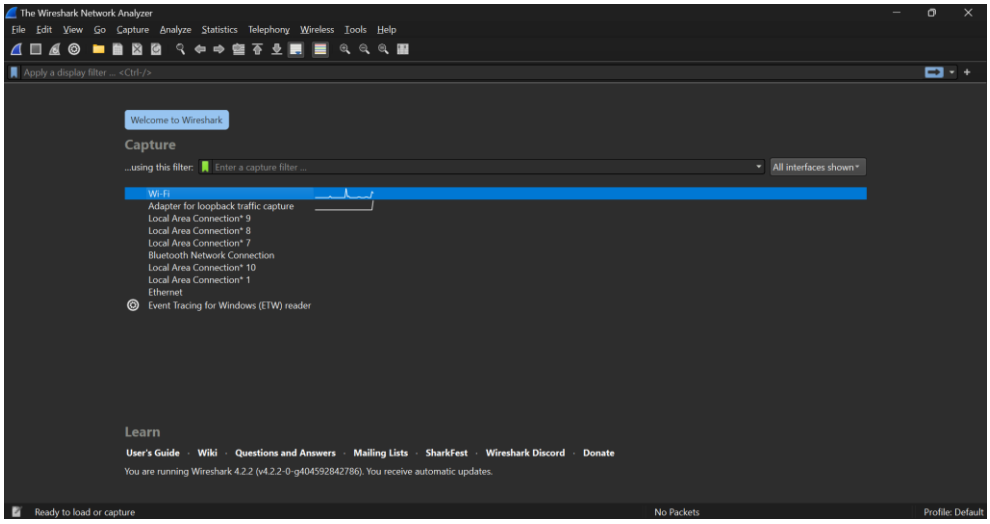


Domain Name System (DNS)

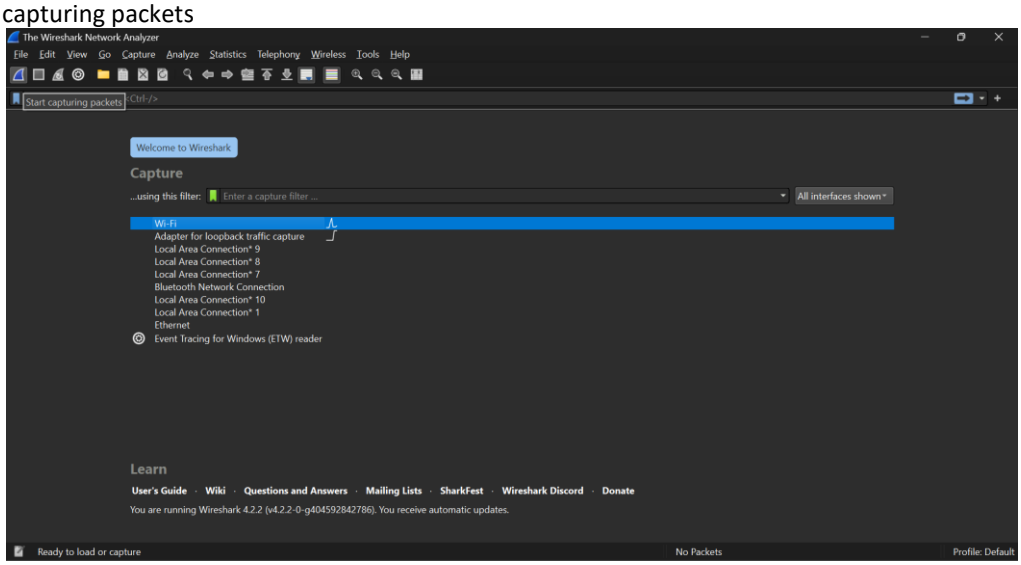
DNS or Domain Name System abbreviated as DNS is a system used to resolve domain names, IP addresses, different servers for e.g., FTP servers, game servers, active directories, etc., and keep their records. Invented by Jon Postel and Paul Mockapetris in 1982, DNS has now become one of the most significant players in the modern-day web world.
DNS actually gives a mapping of the hostname of a network and its address. It has proved to ease human life manifold when one looks at its working and the service it offers. It helps users by translating the domain names into IP addresses, allowing them to surf the web without memorizing such complex IP codes. Coming on to Wireshark, which is an open-source packet analyzer and has been widely in use since its inception in the web world, to analyze packets received or sent in a network. We can use Wireshark to segment the DNS system and get a detailed look at it. The default port for DNS traffic in Wireshark is 53, and the protocol is UDP (User Datagram Protocol).

Wireshark Walk through task for DNS:

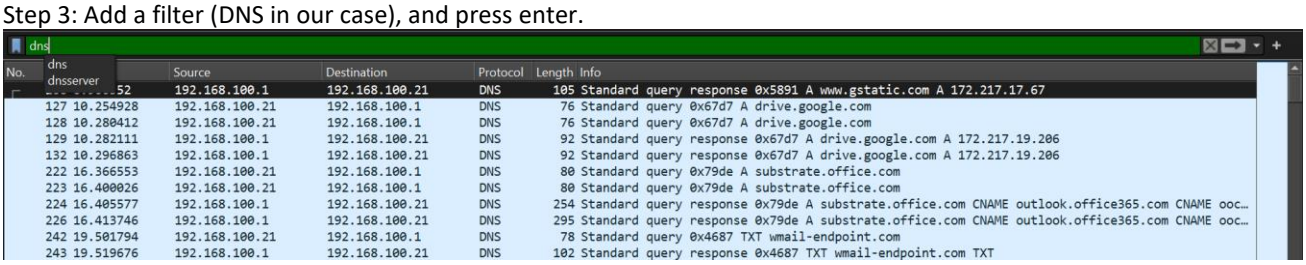
Step 1: Choose the network



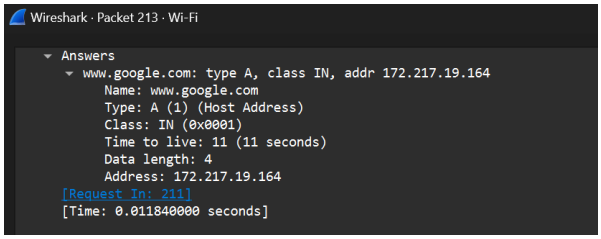
Step 2: Start capturing packets



Step 3: Add a filter (DNS in our case), and press enter.



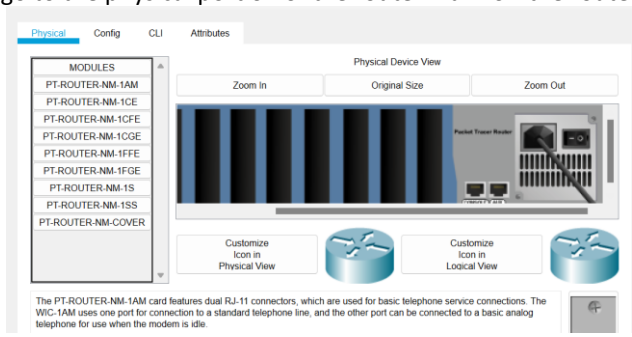
Step 4: Once entered the filter, observe the fields.



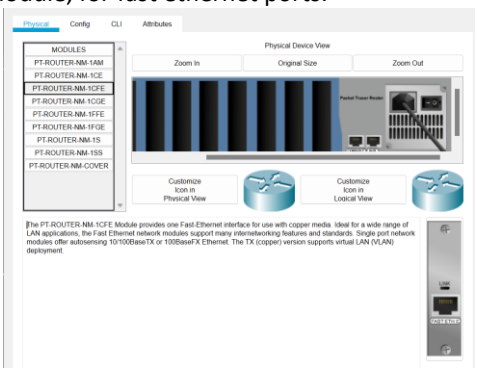
Packet Tracer Walk through task for DNS

Step 1: Select the empty router, and place it on your work area

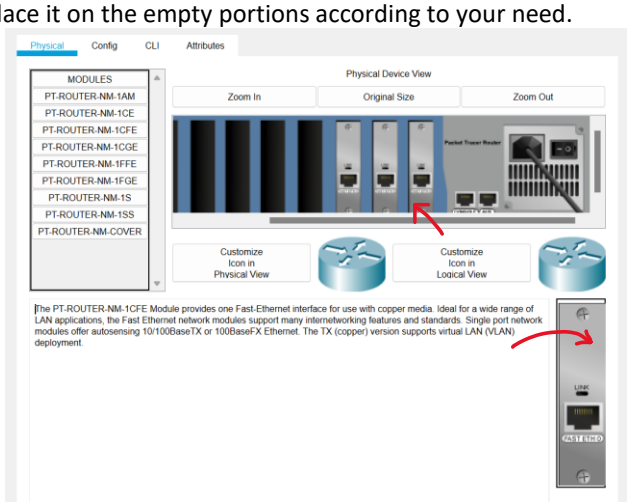
Step 1: click on the router and go to the physical portion of the router. Turn off the router.



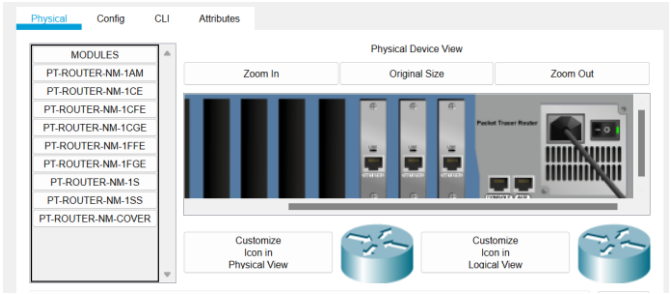
Step 2: Select PT-ROUTER-NM-1CFE Module, for fast ethernet ports.



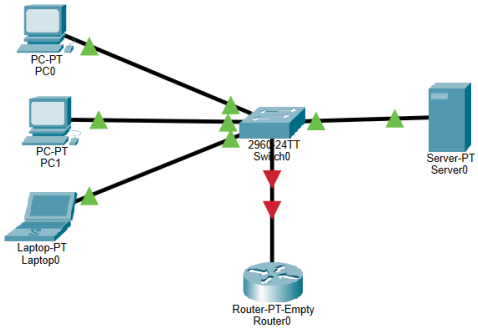
Step 3: Grag the module and place it on the empty portions according to your need.



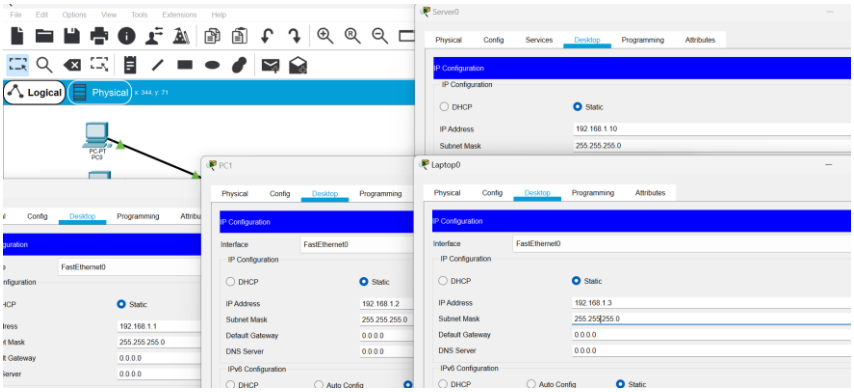
Step 4: Once filled, turn on the router.



Step 5: Once done with the router, construct this topology



Step 6: assign ip addresses to the systems, the server, and the router.
Ip address of pc0: 192.168.1.1
Ip address of pc1: 192.168.1.2
Ip address of Laptop0: 192.168.1.3
Ip address of Server0: 192.168.1.10
Ip address of Router0: 192.168.1.100



Step 7: For Ip address of router follow the following steps:

Step 7.1: Go to CLI portion of the router.
Step 7.2: Enter 'en' for
Step 7.3: Enter 'config t' for

```
Router>
Router>en
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
```

Step 7.4: Once in the configuration terminal, enter the interface you want to configure. In our case Fa0/0 is attached to the switch, hence enter 'interface FastEthernet0/0' and enter.

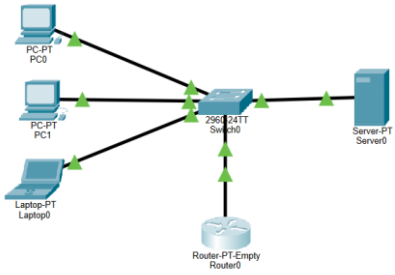
```
Router(config)#interface FastEthernet0/0
Router(config-if)#
```

Step 7.5: Enter the ip address and the subnetmask to the interface. Use this command: 'ip address' ipAddressOfYourInterface Subnetmask, and press enter as shown in the image below.

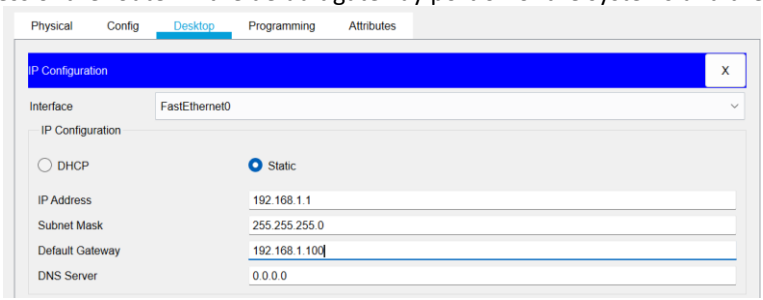
```
Router(config-if)#interface FastEthernet0/0
Router(config-if)#ip address 192.168.1.100 255.255.255.0
Router(config-if)#
```

Step 7.6: enter 'no shutdown' or 'no shut' to turn on that specific interface. This will turn on the port and the the connection will show green light rather than red.

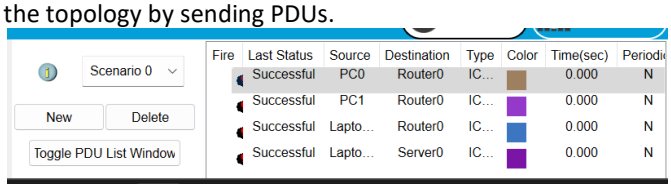
```
Router(config-if)#interface FastEthernet0/0
Router(config-if)#ip address 192.168.1.100 255.255.255.0
Router(config-if)#
Router(config-if)#no shutdown
Router(config-if)#
```



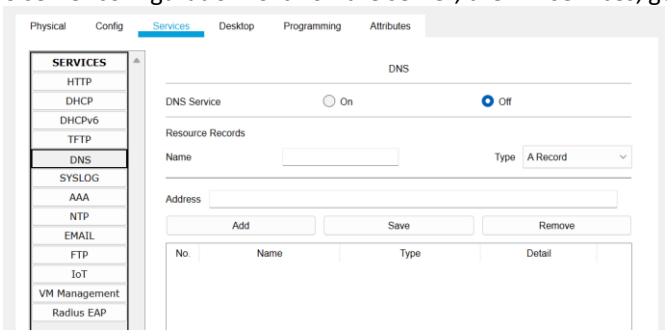
Step 8: Enter the ip address of the router in the default gateway portion of the systems and the server attached.



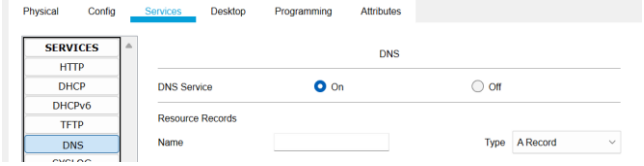
Step 9: Check the working of the topology by sending PDUs.



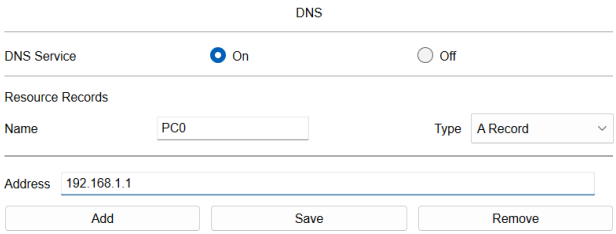
Step 10: Moving on to the DNS server configuration. Click on the server, then in services, go to the DNS portion.



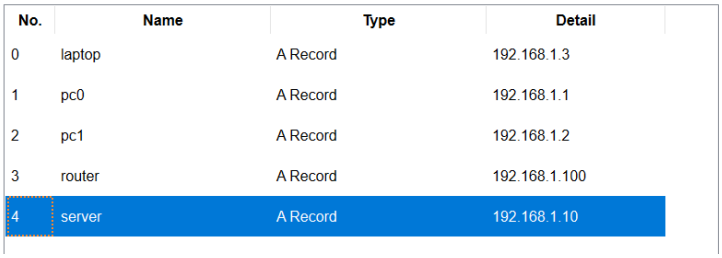
Step 11: Turn on the service.



Step 12: Enter the Name of the system and the ip address assigned to the system, and press add.

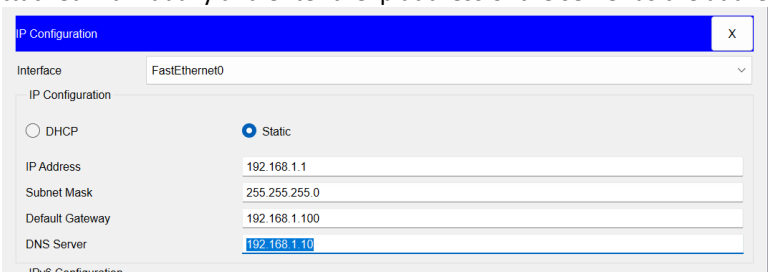


Step 13: Once added, selected that specific entered address and save it.
Step 14: The record will look something like this:

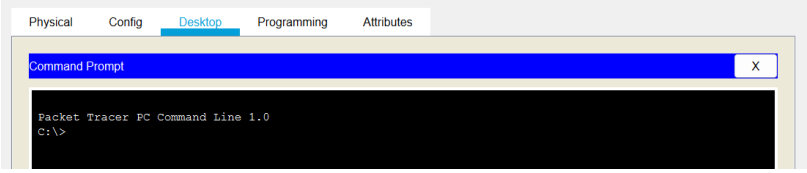


No.	Name	Type	Detail
0	laptop	A Record	192.168.1.3
1	pc0	A Record	192.168.1.1
2	pc1	A Record	192.168.1.2
3	router	A Record	192.168.1.100
4	server	A Record	192.168.1.10

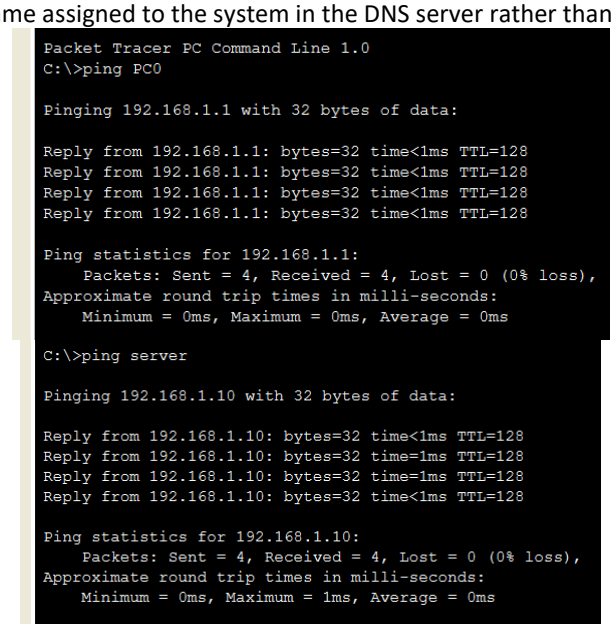
Step 15: Go to systems attached individually and enter the ip address of the server as the address of the DNS server.



Step 16: Once done, go to any one of the systems and to the command prompt.



Step 17: Enter 'ping' and the name assigned to the system in the DNS server rather than the ip address.



```
Packet Tracer PC Command Line 1.0
C:\>ping PC0

Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.1.1: bytes=32 time<1ms TTL=128
Reply from 192.168.1.1: bytes=32 time<1ms TTL=128
Reply from 192.168.1.1: bytes=32 time<1ms TTL=128
Reply from 192.168.1.1: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping server

Pinging 192.168.1.10 with 32 bytes of data:

Reply from 192.168.1.10: bytes=32 time<1ms TTL=128
Reply from 192.168.1.10: bytes=32 time=1ms TTL=128
Reply from 192.168.1.10: bytes=32 time=1ms TTL=128
Reply from 192.168.1.10: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Step 18: This is how you access DNS server.

National University of Computer & Emerging Sciences (NUCES), Islamabad

Practice Tasks

Task 1:

Answer the following questions for the assigned website:

1. Is your browser running HTTP version 1.0 or 1.1?
2. What languages (if any) does your browser indicate that it can accept to the server?
3. What is the IP address of your computer and the website?
4. What is the status code returned from the server to your browser?
5. When was the HTML file that you are retrieving last modified at the server?
6. How many bytes of content are being returned to your browser?
7. By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.
8. Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?
9. What is the status code and phrase in the response?
10. When your browser's sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

For this specific question, you are required to answer the question on a word or pdf file. Attach the screenshot of the Packets captured screen from Wireshark, and then answer the questions.

Task 2:

Answer the following questions by observing packets on wireshark

1. Does the DNS request packet indicate the use of any specific DNS record type (e.g., A, AAAA, MX)?
2. What is the domain name being queried in the DNS request packet?
3. Is the DNS query using UDP or TCP as the transport protocol?
4. What is the IP address of the DNS server that provided the response?
5. Does the DNS response packet include the requested IP address(es) for the queried domain?
6. Are there any DNS response packets indicating that the queried domain does not exist (NXDOMAIN)?
7. Do you observe any DNS packets with the "Truncated" flag set, indicating a truncated response?
8. What is the Time-to-Live (TTL) value in the DNS response packet for the resolved IP address?

You are required to answer the question on a word or pdf file, after you are done with Task 1. Attach the screenshot of the Packets captured screen from Wireshark, and then answer the questions.

Task 3:

Construct a client server model that have 9 hosts, three of them are computers, and other six host will be laptops, in which further connected to 2 switches, assign them IP address and subnet masks dynamically from the server. Moreover, access the HTTP server from the client and show the results. You are required to make a single .html file and import it to the server and access it from the client.

Task 4:

Construct a client server model in bus topology. Use 7 systems attached to a separate switch. Attach a server in such a way that it provides DNS services to all. Send PDUs to check your topology.

Submission Guidelines :

1. Do not zip your tasks, upload your tasks separately on GCR with naming convention: rollNumber_TaskNumber.pkt, along with the screenshot of the topology with the successful message.
2. Pledgerism will result in ZERO marks.
3. No late submission will be marked.