

NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCE

Computer Network Lab (CL-3001)

Lab Session 04

OBJECTIVE: To understand Application Layer Protocol in Cisco Packet Tracer

DYNAMIC HOST CONFIGURATION PROTOCOL (DHCP)

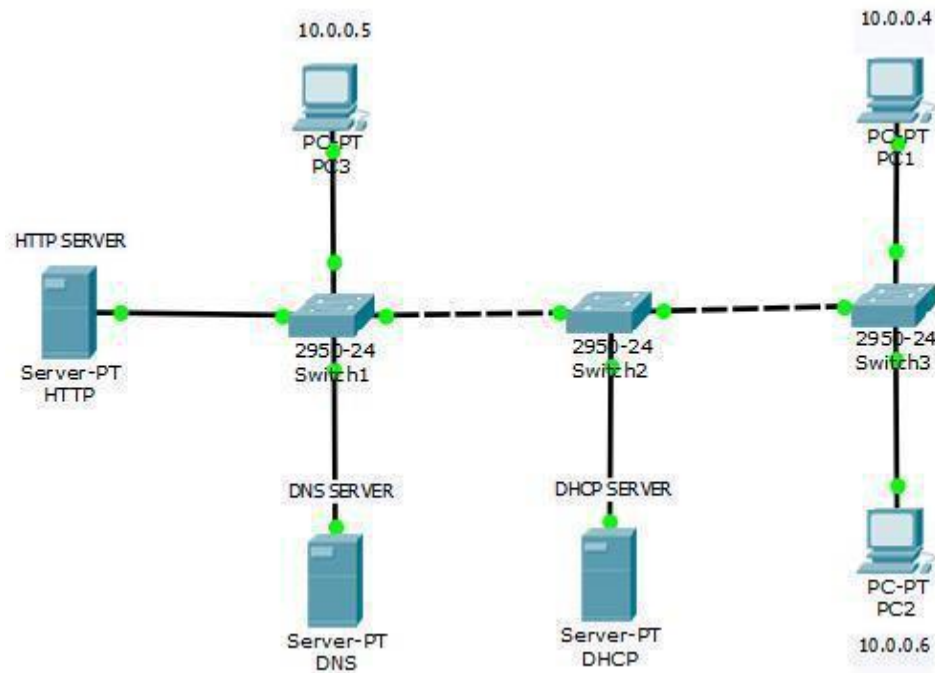
An IP address can be defined as a unique numeric identifier (address) that is assigned to each computer operating in a TCP/IP based network. Manually configuring computers with IP addresses and other TCP/IP configuration parameters is not an intricate task. However, manually configuring thousands of workstations with unique IP addresses would be a time consuming, and cumbersome experience. When you manually assign IP addresses, you increase the risk of duplicating IP address assignments, configuring the incorrect subnet masks, and incorrectly configuring other TCP/IP configuration parameters. This is where the Dynamic Host Configuration Protocol (DHCP) becomes important. The Dynamic Host Configuration Protocol (DHCP) is a service that does the above mentioned tasks for administrators, thereby saving simplifying the administration of IP addressing in TCP/IP based networks. TCP/IP configuration was basically a manual process before the DHCP protocol was introduced. One of the main disadvantages of manually assigning IP addresses to hundreds of computers is that it could result in the assigned IP addresses not being unique.

You should only use manual address assignment under these circumstances:

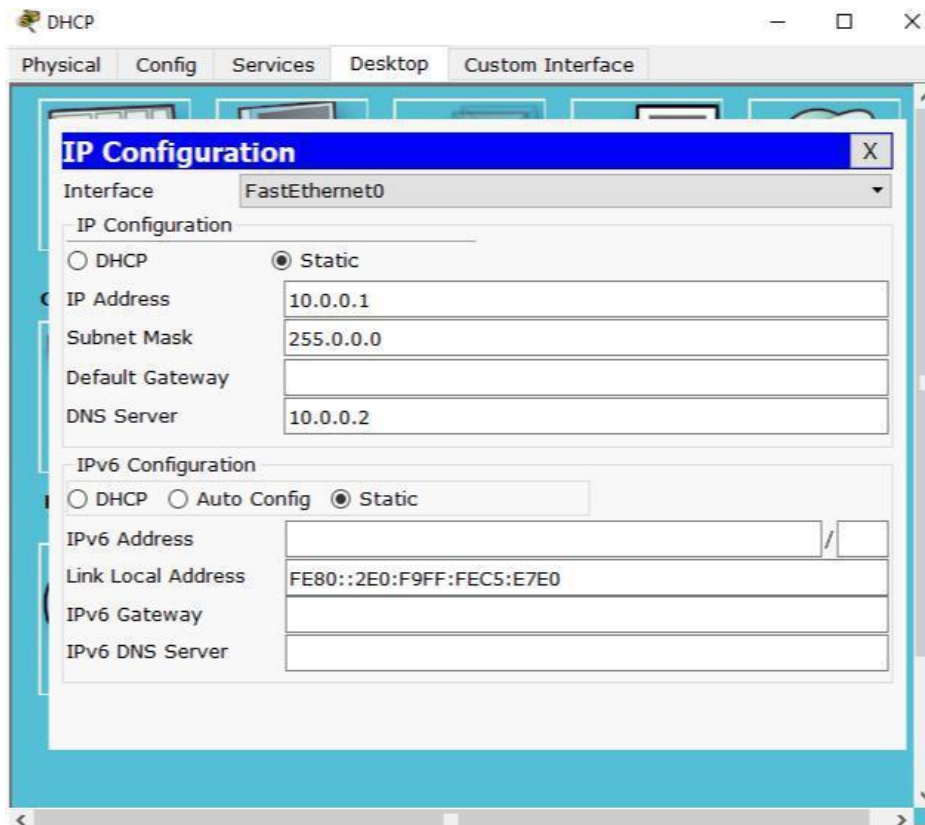
- When there are no configured DHCP servers on the network and the network has multiple network segments.
- When you are configuring a computer as a DHCP server, you assign that computer a static IP address.
- When you configure computers as important network servers such as domain controllers, or DNS servers; you manually assign the IP address to these computers.

DHCP functions at the application layer of the TCP/IP protocol stack. One of the primary tasks of the protocol is to automatically assign IP addresses to DHCP clients. A server running the DHCP service is called a DHCP server. The DHCP protocol automates the configuration of TCP/IP clients because IP addressing occurs through the system. You can configure a server as a DHCP server so that the DHCP server can automatically assign IP addresses to DHCP clients, and with no manual intervention.

Now moving on to the implementation, first create the following:



Click on the DHCP server, and assign IP as follows:



Then go to Services Tab and perform the following:

Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP
server...	0.0.0.0	10.0.0.2	10.0.0.4	255.0.0.0	512	0.0.0.0
server...	0.0.0.0	0.0.0.0	10.0.0.0	255.0.0.0	512	0.0.0.0

Here, we can clearly see that “Start IP Address” field has value **10.0.0.3** since 10.0.0.1 and 10.0.0.2 and 10.0.0.3 are reserved for DHCP, DNS and HTTP servers respectively.

Before assigning the IPs to PCs, we’ll be configuring those servers as well.

DOMAIN NAME SYSTEM (DNS)

The Domain Name System (DNS) is a hierarchical distributed naming system for computers, services, or any resource connected to the Internet or a private network. It associates information with domain names assigned to each of the participating entities. Most prominently, it translates easily memorized domain names to the numerical IP addresses needed for the purpose of locating computer services and devices worldwide. The Domain Name System is an essential component of the functionality of the Internet.

An often-used analogy to explain the Domain Name System is that it serves as the phone book for the Internet by translating human-friendly computer hostnames into IP addresses. For example, the domain name `www.example.com` translates to the addresses `93.184.216.119` (IPv4) and `2606:2800:220:6d:26bf:1447:1097:aa7` (IPv6). Unlike a phone book, the DNS can be quickly updated, allowing a service's location on the network to change without affecting the end users, who continue to use the same host name. Users take advantage of this when they use meaningful Uniform Resource Locators (URLs), and e-mail addresses without having to know how the computer actually locates the services.

The Domain Name System distributes the responsibility of assigning domain names and mapping those names to IP addresses by designating authoritative name servers for each domain. Authoritative name servers are assigned to be responsible for their supported domains, and may delegate authority over sub domains to other name servers. This mechanism provides distributed and fault tolerant service and was designed to avoid the need for a single central database. Some common DNS record types are:

A RECORD:

The A record is one of the most commonly used record types in any DNS system. An A record is actually an address record, which means it maps a fully qualified domain name (FQDN) to an IP address. For example, an A record is used to point a domain name, such as "google.com", to the IP address of Google's hosting server, "74.125.224.147". This allows the end user to type in a human- readable domain, while the computer can continue working with numbers. The name in the A record is the host for your domain, and the domain name is automatically attached to your name.

CNAME record:

Canonical name records, or CNAME records, are often called alias records because they map an alias to the canonical name. When a name server finds a CNAME record, it replaces the name with the canonical name and looks up the new name. This allows pointing multiple systems to one IP without assigning an A record to each host name. It means that if you decide to change your IP address, you will only have to change one A record.

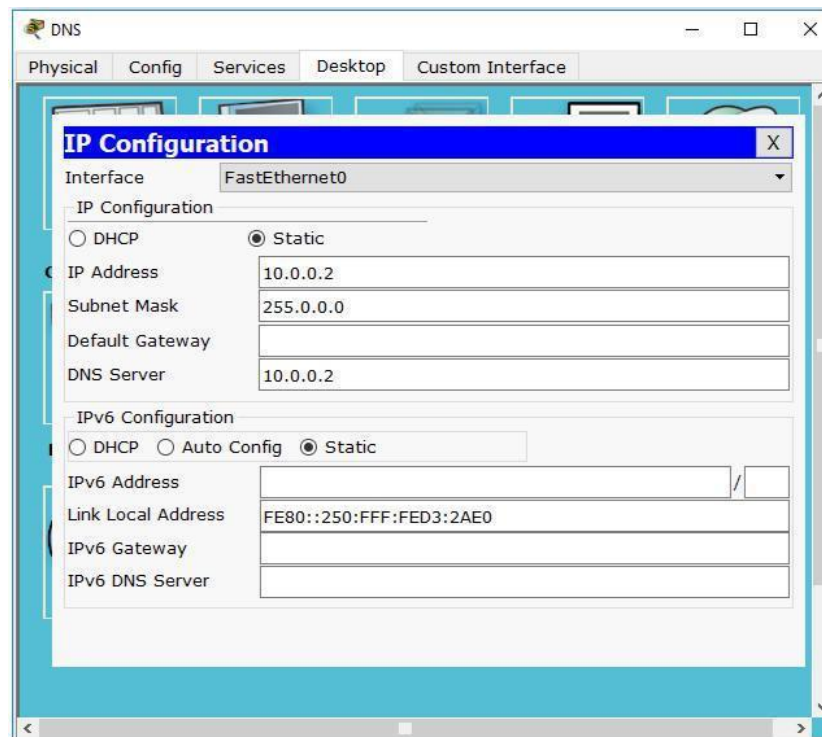
NS record:

An NS record identifies which DNS server is authoritative for a particular zone. The "NS" stands for "name server". NS records that do not exist on the apex of a domain are primarily used for splitting up the management of records on sub-domains.

SOA record:

The SOA or Start of Authority record for a domain stores information about the name of the server that supplies the data for the zone, the administrator of the zone and the current version of the data. It also provides information about the number of seconds a secondary name server should wait before checking for updates or before retrying a failed zone transfer.

Now back to implementation part: Click on DNS Server and assign IP as follows:



Then go to Services Tab and perform the following:

The screenshot shows the 'DNS' configuration window with the 'Services' tab selected. On the left, a 'SERVICES' list includes HTTP, DHCP, DHCPv6, TFTP, DNS, SYSLOG, AAA, NTP, EMAIL, and FTP. The main area is titled 'DNS' and contains the following fields and controls:

- DNS Service:** A radio button interface with 'On' selected and 'Off' unselected.
- Resource Records:** A section for adding records.
 - Name:** A text field containing 'cnlab'.
 - Type:** A dropdown menu set to 'A Record'.
 - Address:** A text field containing '10.0.0.3'.
 - Buttons:** 'Add', 'Save', and 'Remove' buttons are located below the address field.
- Record Table:** A table with columns 'No.', 'Name', 'Type', and 'Detail'. It contains two entries:

No.	Name	Type	Detail
0	cn	CNAME	cnlab
1	cnlab	A Record	10.0.0.3
- DNS Cache:** A button located at the bottom of the configuration area.

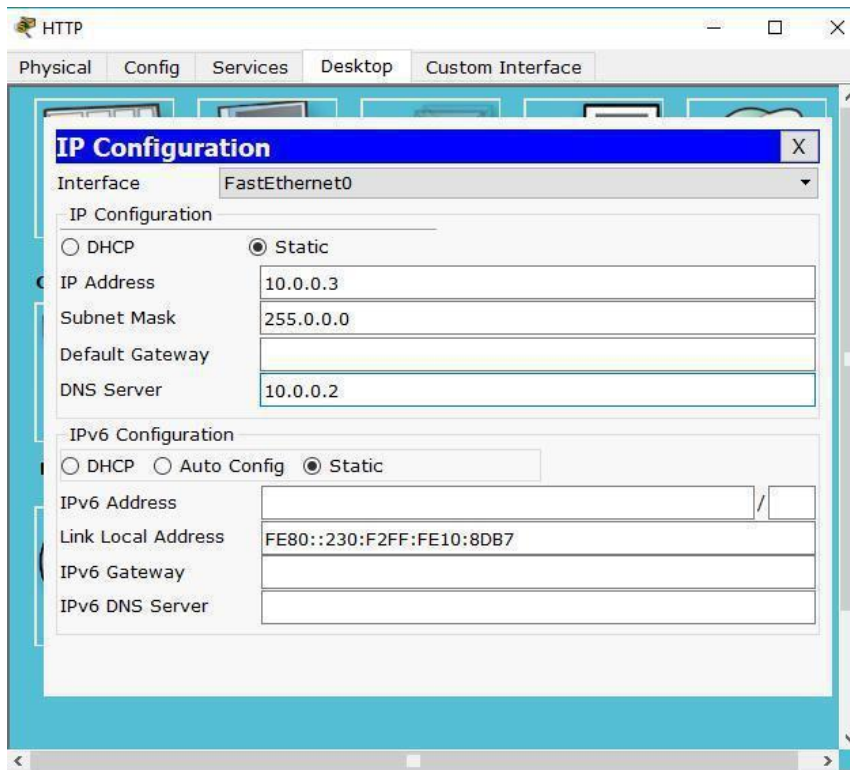
The “Address” field has the IP address which we’ll assign to HTTP Server later.

This screenshot shows the same 'DNS' configuration window, but with the 'Name' field set to 'cn' and the 'Type' dropdown set to 'CNAME'. The 'Host Name' field now contains 'cnlab'. The 'Record Table' has been updated to reflect these changes:

No.	Name	Type	Detail
0	cn	CNAME	cnlab
1	cnlab	A Record	10.0.0.3

The 'Add', 'Save', and 'Remove' buttons remain visible below the table.

Now click on the HTTP Server and assign IP address as follows:



Finally, we will now assign the IPs to PC1, PC2 and PC3 via DHCP and analyze it. For that, we need to perform the following:

Go to Simulation Mode, Click on PC1, then go to Desktop Tab, Select IP Configuration and select DHCP.

Do the same for PC2 and PC3.

Vis.	Time(sec)	Last Device	At Device	Type	Info
	1.513	Switch1	HTTP	DHCP	
	1.513	Switch3	PC1	DHCP	
	1.513	Switch3	PC2	DHCP	
	1.514	PC2	Switch3	DHCP	
	1.515	Switch3	Switch2	DHCP	
	1.515	Switch3	PC1	DHCP	
	1.516	Switch2	Switch1	DHCP	
	1.516	Switch2	DHCP	DHCP	
	1.517	Switch1	DNS	DHCP	

Reset Simulation ☒ Constant Delay Captured to: * 1.519 s

PDU Information at Device: DHCP

OSI Model Inbound PDU Details Outbound PDU Details

At Device: DHCP
Source: DHCP
Destination: Broadcast

In Layers	Out Layers
Layer 7: DHCP Frame Server: 10.0.0.1, Client: 0.0.0.0	Layer 7: DHCP Frame Server: 10.0.0.1, Client: 0.0.0.0
Layer 6	Layer 6
Layer 5	Layer 5
Layer 4: UDP Src Port: 68, Dst Port: 67	Layer 4: UDP Src Port: 67, Dst Port: 68
Layer 3: IP Header Src. IP: 0.0.0.0, Dest. IP: 255.255.255.255	Layer 3: IP Header Src. IP: 10.0.0.1, Dest. IP: 255.255.255.255
Layer 2: Ethernet II Header 00D0.FF58.C3A9 >> FFFF.FFFF.FFFF	Layer 2: Ethernet II Header 00E0.F9C5.E7E0 >> FFFF.FFFF.FFFF
Layer 1: Port FastEthernet0	Layer 1: Port(s): FastEthernet0

1. FastEthernet0 receives the frame.

Challenge Me

<< Previous Layer

Next Layer >>

Next, we'll be analyzing DNS packets in simulation, for that, Click on PC1, in Desktop Tab select Web Browser and then enter cnlab, cn, or its address in the URL provided and observe the packet transmission in simulation mode. You should get to see something like this:

The image displays a network simulation environment. At the top, a network topology is shown with three switches (2950-24 Switch1, Switch2, and Switch3) connected in a line. Various servers (HTTP, DNS, DHCP) and PCs (PC1, PC2, PC3) are connected to the switches. Below the topology, a PC1 window is open, showing a Web Browser with the URL 'http://cnlab'. To the right, a Simulation Panel is visible, containing an Event List with a single entry for a DNS packet at 0.000 seconds. The panel also includes Play Controls (Back, Auto Capture / Play, Capture / Forward) and Event List Filters.

Click on “Capture/Forward” or “Auto Capture/Play” to see how it goes and note down your observation.

The screenshot displays a network topology in Cisco Packet Tracer. The topology includes an HTTP SERVER (Server-PT HTTP) connected to Switch1 (2950-24), which is connected to Switch2 (2950-24), which is connected to Switch3 (2950-24). Switch1 is also connected to a DNS SERVER (Server-PT DNS) and a PC-PT PC3 (10.0.0.5). Switch2 is connected to a DHCP SERVER (Server-PT DHCP). Switch3 is connected to a PC-PT PC2 (10.0.0.6). The Simulation Panel on the right shows the Event List with the following data:

Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.020	Switch2	Switch1	HTTP	
	0.020	Switch1	HTTP	TCP	
	0.021	Switch1	HTTP	HTTP	
	0.022	Switch1	Switch1	HTTP	
	0.023	Switch1	Switch2	HTTP	
	0.024	Switch2	Switch3	HTTP	
	0.025	--	PC1	TCP	
	0.025	Switch3	PC1	HTTP	
	0.025	--	PC1	TCP	

The Play Controls section shows the "Auto Capture / Play" button is active. The Event List Filters - Visible Events section lists various protocols and services.

The screenshot displays the Cisco Packet Tracer Student interface. The network topology is the same as in the previous screenshot. The Simulation Panel on the right shows the Event List with the following data:

Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.000	--	PC1	DNS	
	0.001	PC1	Switch3	DNS	
	0.002	Switch3	Switch2	DNS	
	0.003	Switch2	Switch1	DNS	
	0.004	Switch1	DNS	DNS	
	0.005	DNS	Switch1	DNS	
	0.006	Switch1	Switch2	DNS	
	0.007	Switch2	Switch3	DNS	
	0.008	--	PC1	TCP	

The Play Controls section shows the "Auto Capture / Play" button is active. The Event List Filters - Visible Events section lists various protocols and services. The bottom status bar shows the time as 00:34:53.411 and the power cycle devices button is active.

In above figure, we can see DNS type in Simulation Panel, Click on any of them and open it to observe its details. We can even see that in OSI model, DNS is present on Layer 7(Application Layer).

PDU Information at Device: DNS

OSI Model Inbound PDU Details Outbound PDU Details

At Device: DNS
Source: PC1
Destination: 10.0.0.2

In Layers

Layer 7: DNS
Layer6
Layer5
Layer 4: UDP Src Port: 1030, Dst Port: 53
Layer 3: IP Header Src. IP: 10.0.0.4, Dest. IP: 10.0.0.2
Layer 2: Ethernet II Header 0090.0CCC.723D >> 0050.0FD3.2AE0
Layer 1: Port FastEthernet0

Out Layers

Layer 7: DNS
Layer6
Layer5
Layer 4: UDP Src Port: 53, Dst Port: 1030
Layer 3: IP Header Src. IP: 10.0.0.2, Dest. IP: 10.0.0.4
Layer 2: Ethernet II Header 0050.0FD3.2AE0 >> 0090.0CCC.723D
Layer 1: Port(s): FastEthernet0

1. FastEthernet0 receives the frame.

Challenge Me << Previous Layer Next Layer >>

Similarly, we can analyze HTTP details as well.

PDU Information at Device: HTTP

OSI Model Inbound PDU Details Outbound PDU Details

At Device: HTTP
Source: PC1
Destination: HTTP CLIENT

In Layers	Out Layers
Layer 7: HTTP	Layer 7: HTTP
Layer 6	Layer 6
Layer 5	Layer 5
Layer 4: TCP Src Port: 1031, Dst Port: 80	Layer 4: TCP Src Port: 80, Dst Port: 1031
Layer 3: IP Header Src. IP: 10.0.0.4, Dest. IP: 10.0.0.3	Layer 3: IP Header Src. IP: 10.0.0.3, Dest. IP: 10.0.0.4
Layer 2: Ethernet II Header 0090.0CCC.723D >> 0030.F210.8DB7	Layer 2: Ethernet II Header 0030.F210.8DB7 >> 0090.0CCC.723D
Layer 1: Port FastEthernet0	Layer 1: Port(s): FastEthernet0

1. FastEthernet0 receives the frame.

Challenge Me << Previous Layer Next Layer >>

DHCP in Wireshark:

As we've done in earlier Wireshark labs, you'll perform a few actions on your computer that will cause DHCP to spring into action, and then use Wireshark to collect and then the packet trace containing DHCP protocol messages.

The first two steps in the DHCP protocol in Figure 4.24 (using the Discover and Offer messages) are optional (in the sense that they need not always be used when, for example, a new IP address is needed, or an existing DHCP address is to be renewed); the Request and ACK messages are not. In order to collect a trace that will contain all four DHCP message types, we'll need to take a few command line actions on a Mac, Linux or PC.

On a PC:

1. In a command-line window enter the following command:

```
> ipconfig /release
```

This command will cause your PC to give up its IP address.

2. Start up Wireshark.

3. In the command-line window enter the following command:

```
> ipconfig /renew
```

This will cause the DHCP protocol to request and receive an IP address and other information from a DHCP server.

4. After waiting for a few seconds, stop Wireshark capture.

After stopping Wireshark capture in step 4, you should take a peek in your Wireshark window to make sure you've actually captured the packets that we're looking for. If you enter "dhcp" into the display filter field (as shown in the light green field in the top left of Figure 1), your screen (on a Mac) should look similar to Figure 1.

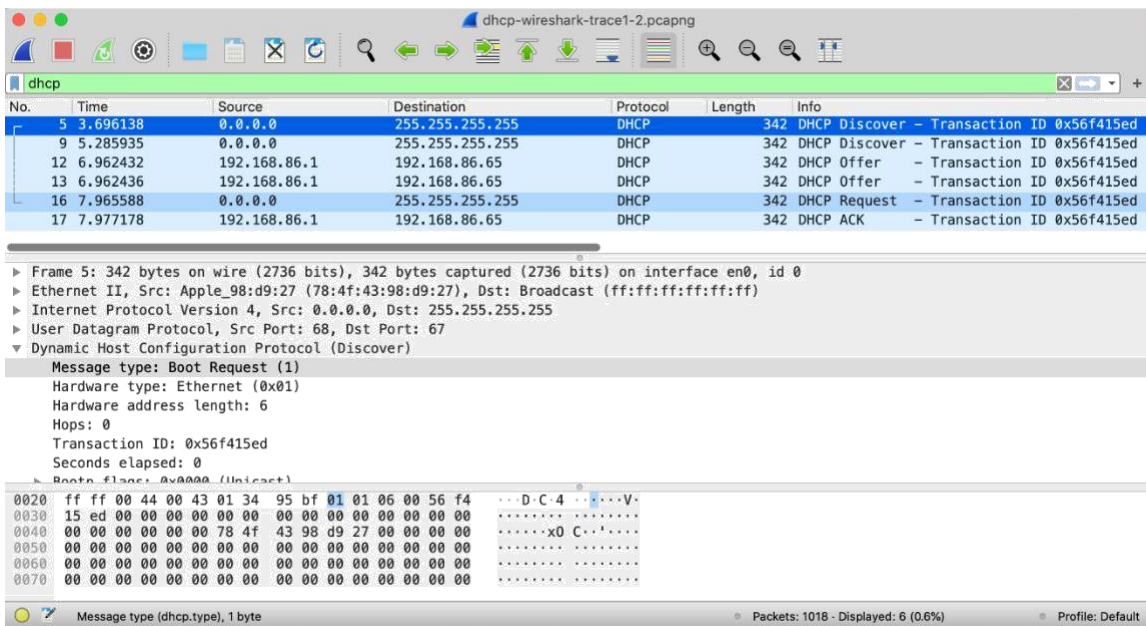


Figure 1: Wireshark display, showing the capture of DHCP Discover, Offer, Request and ACK messages

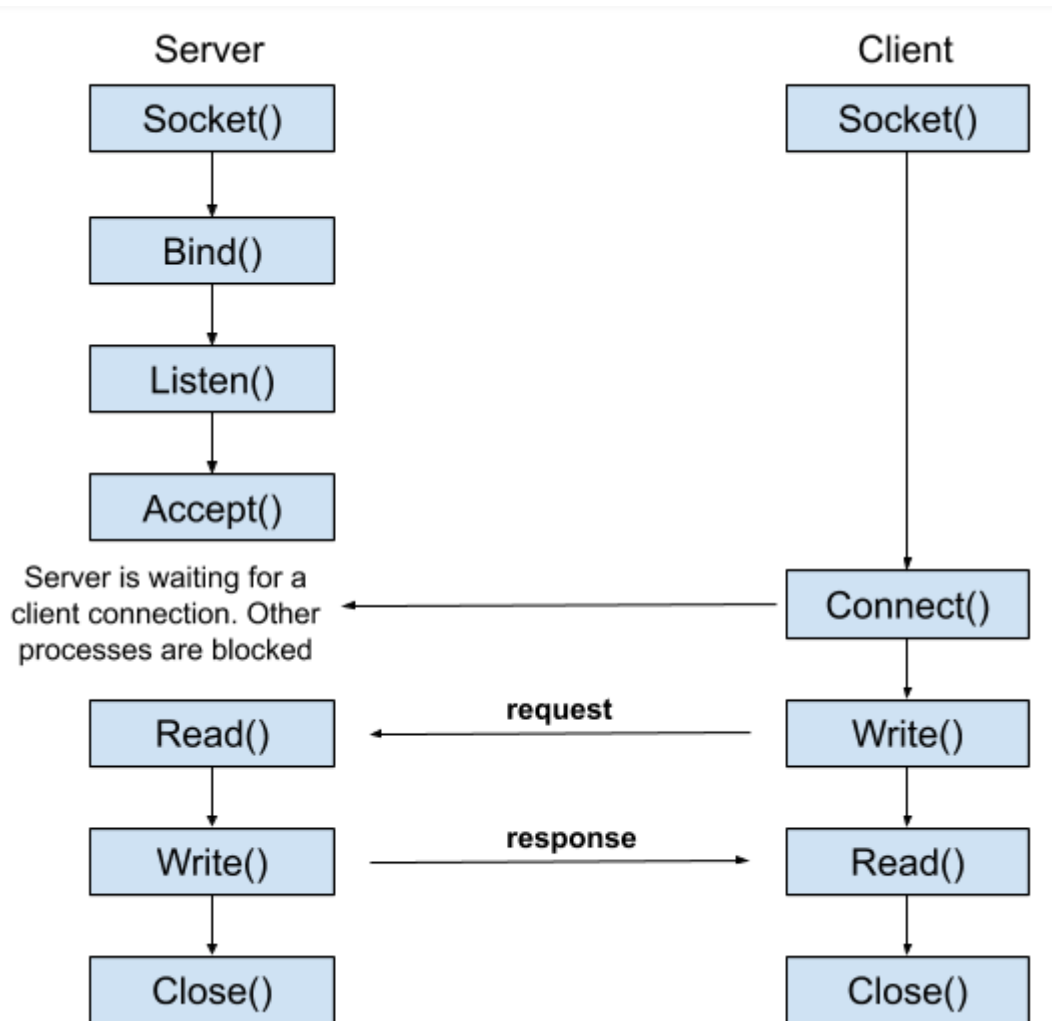
Answer the following questions

Let's start by looking at the DHCP Discover message. Locate the IP datagram containing the first Discover message in your trace.

1. Is this DHCP Discover message sent out using UDP or TCP as the underlying transport protocol?
2. What is the source IP address used in the IP datagram containing the Discover message? Is there anything special about this address? Explain.
3. What is the destination IP address used in the datagram containing the Discover message. Is there anything special about this address? Explain.
4. What is the value in the transaction ID field of this DHCP Discover message?
5. Now inspect the options field in the DHCP Discover message. What are five pieces of information (beyond an IP address) that the client is suggesting or requesting to receive from the DHCP server as part of this DHCP transaction?

Socket Programming

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server



What are the steps used for socket programming?

The steps involved in establishing a TCP socket on the server side are as follows:

- Create a socket with the `socket()` function;
- Bind the socket to an address using the `bind()` function;
- Listen for connections with the `listen()` function;
- Accept a connection with the `accept()` function system call

Why are sockets useful?

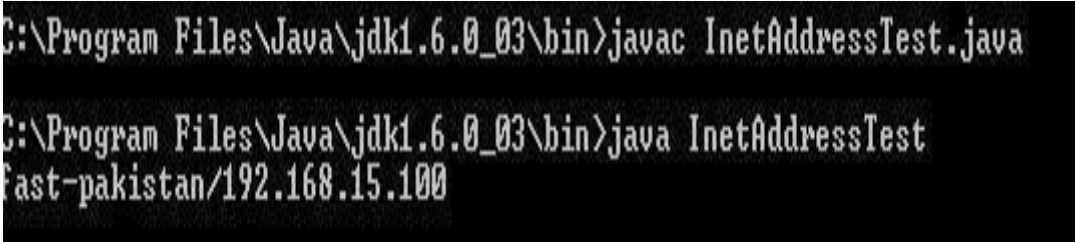
Sockets are useful for both stand-alone and network applications. Sockets allow you to exchange information between processes on the same machine or across a network,

distribute work to the most efficient machine, and they easily allow access to centralized data.

Example 1: How to get IP address in java using InetAddress

import java.net.UnknownHostException;
import java.net.InetAddress;
class InetAddressTest
{
public static void main(String[] args)
throws UnknownHostException
{
InetAddress Address=InetAddress.getLocalHost();
System.out.println(Address);
}
}

Output



```
C:\Program Files\Java\jdk1.6.0_03\bin>javac InetAddressTest.java
C:\Program Files\Java\jdk1.6.0_03\bin>java InetAddressTest
fast-pakistan/192.168.15.100
```

Example 2:How to get Host Address

```
import java.net.UnknownHostException;
import java.net.InetAddress;
public class IPTest {
    public static void main(String args[]) throws UnknownHostException {
        InetAddress addr = InetAddress.getLocalHost();
        String ipAddress = addr.getHostAddress();
        System.out.println("IP address of localhost from Java Program: " + ipAddress);
        //Hostname
        String hostname = addr.getHostName();
        System.out.println("Name of hostname : " + hostname);
    }
}
```

Output

```
C:\Program Files\Java\jdk1.6.0_03\bin>javac IPTest.java
C:\Program Files\Java\jdk1.6.0_03\bin>java IPTest
IP address of localhost from Java Program: 192.168.15.100
Name of hostname : fast-pakistan
```

EXERCISE:

- 1) What Does A Socket Consists Of?
- 2) How To Make A Socket A Listen-only Connection Endpoint - Listen()?
- 3) Design Client Server Application.
- 4) Design the following networks shown in figure 1 & 2 below using DHCP.
Observe the PDU information and write down OSI layers packet information.

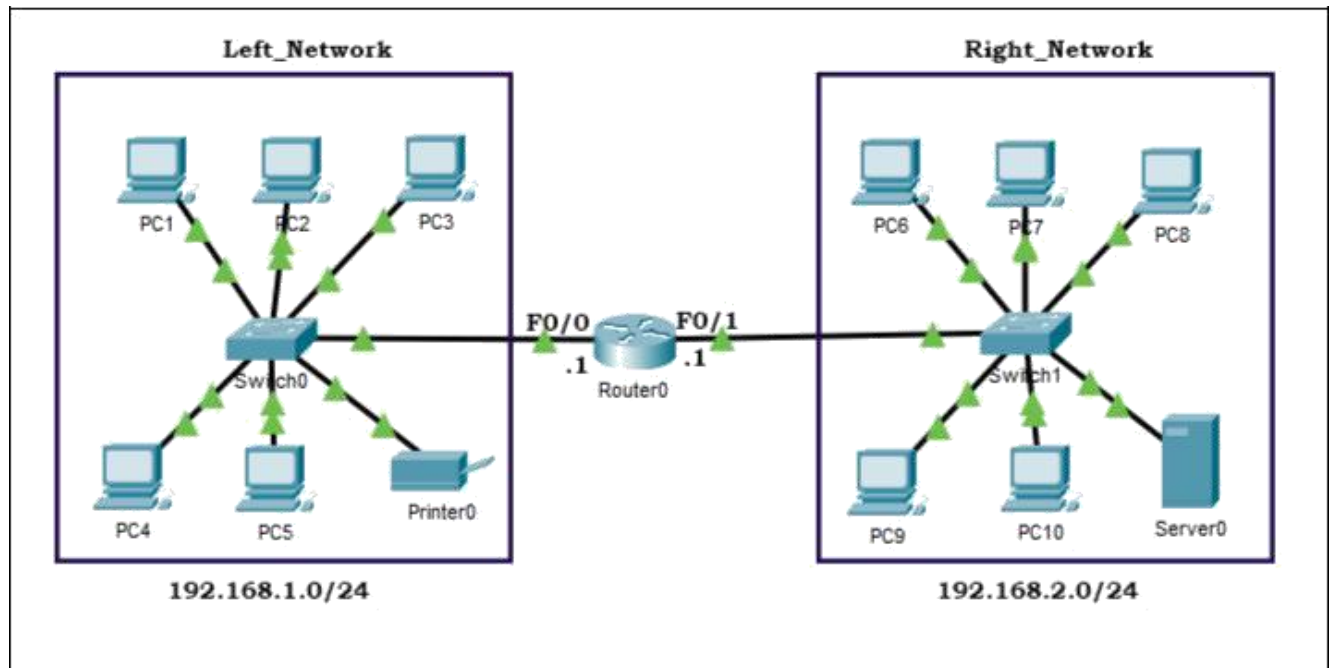


Figure 1

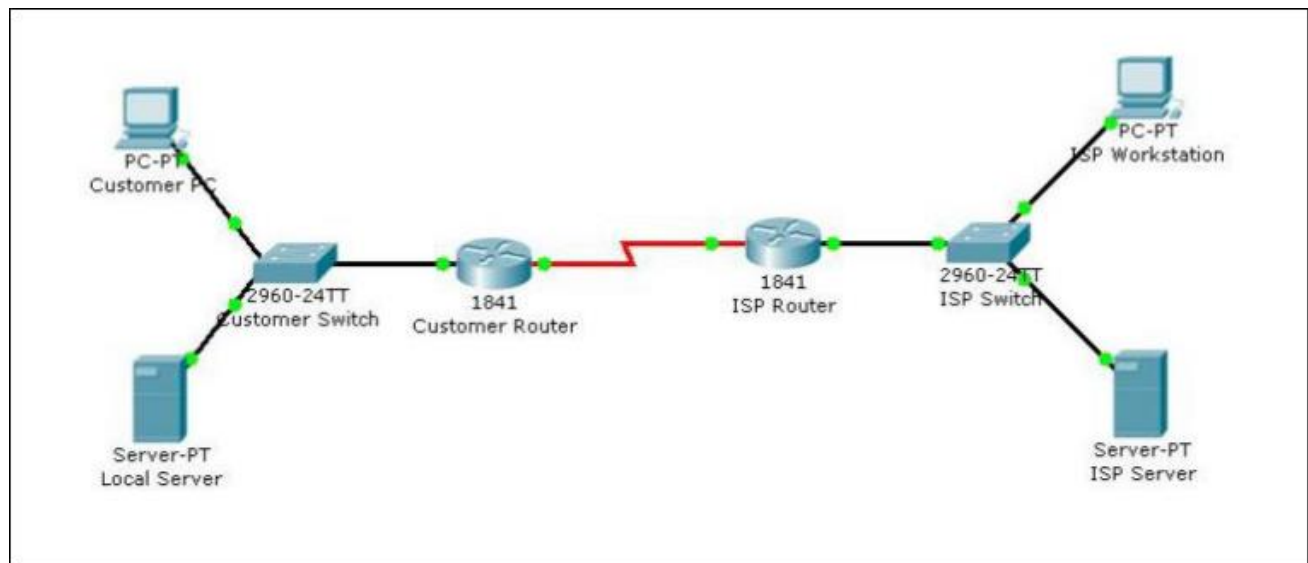


Figure 2