

**NATIONAL UNIVERSITY OF COMPUTER &  
EMERGING SCIENCE**  
**Computer Networks Lab (CL307)**  
**Lab Session 06**

**Application Layer Protocol**

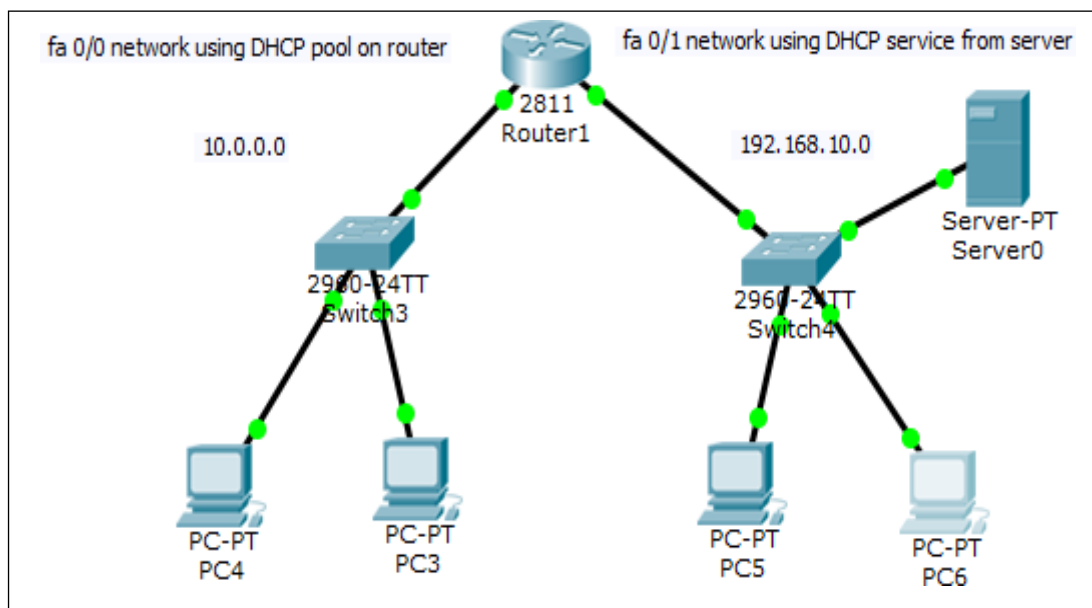
**DHCP & DNS in Cisco Packet Tracer**

**DYNAMIC HOST CONFIGURATION PROTOCOL**

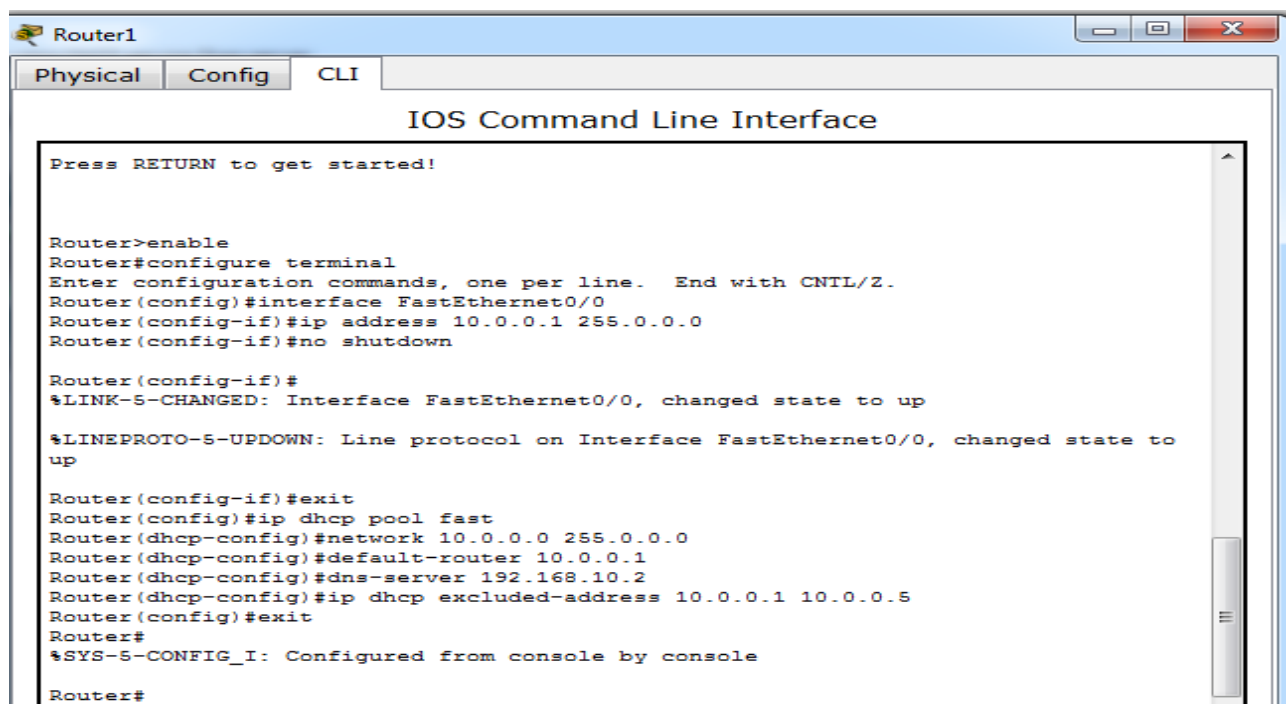
The Dynamic Host Configuration Protocol is used by computers for requesting Internet Protocol parameters, such as an IP address from a network server. The protocol operates based on the client-server model. DHCP is very common in all modern networks ranging in size from home networks to large campus networks and regional Internet service provider networks. Most residential network routers receive a globally unique IP address within the provider network. Within a local network, DHCP assigns a local IP address to devices connected to the local network.

When a computer or other networked device connects to a network, its DHCP client software in the operating system sends a broadcast query requesting necessary information. Any DHCP server on the network may service the request. The DHCP server manages a pool of IP addresses and information about client configuration parameters such as default gateway, domain name, the name servers, time servers. On receiving a request, the server may respond with specific information for each client, as previously configured by an administrator, or with a specific address and any other information valid for the entire network, and the time period for which the allocation (*lease*) is valid. A host typically queries for this information immediately after booting, and periodically thereafter before the expiration of the information. When an assignment is refreshed by the client computer, it initially requests the same parameter values, but may be assigned a new address from the server, based on the assignment policies set by administrators.

We can use DHCP service from router as well as from Server.



## Now configuring network on Fa 0/0.



```
Router1
Physical Config CLI
IOS Command Line Interface

Press RETURN to get started!

Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown

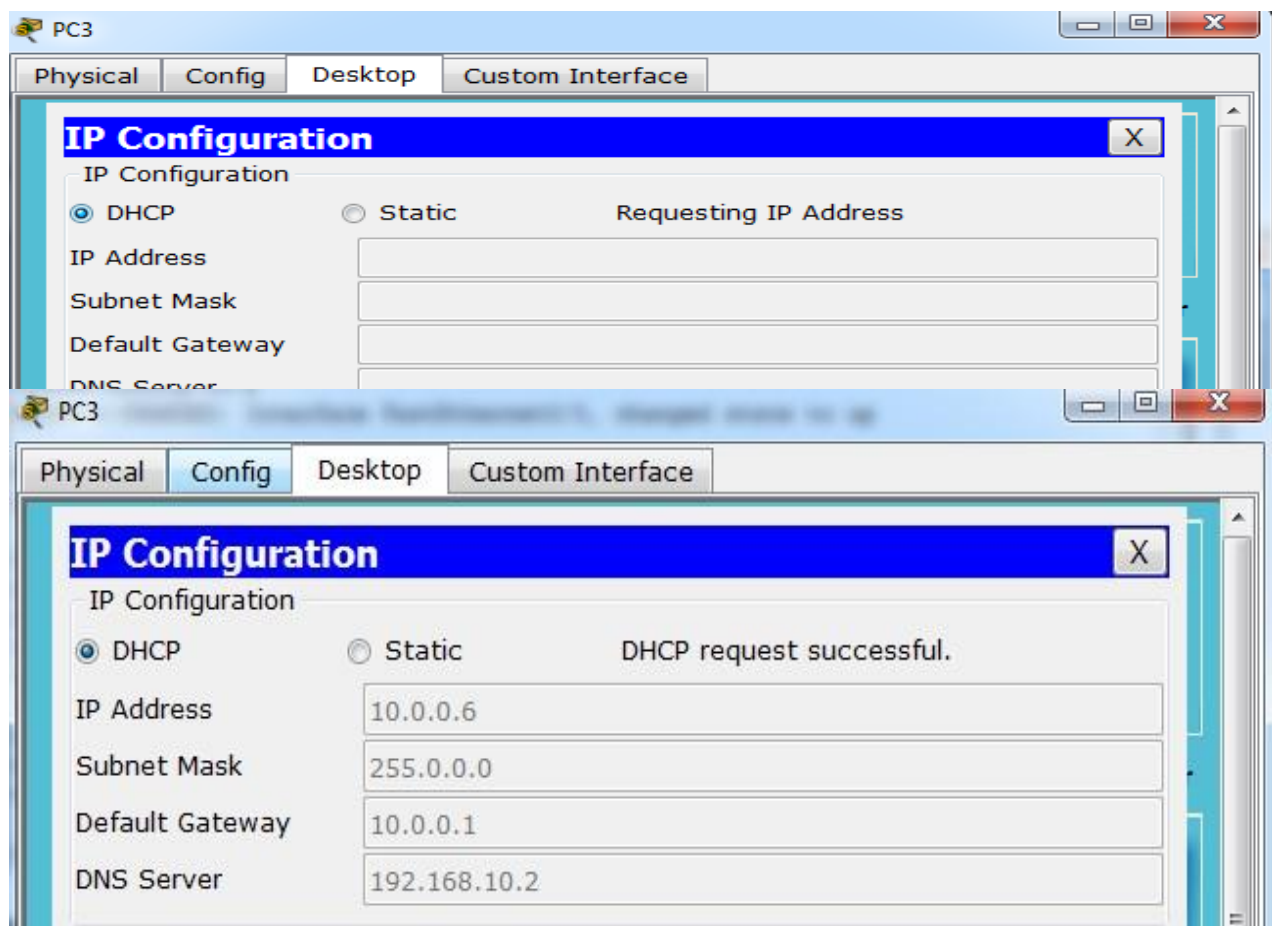
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

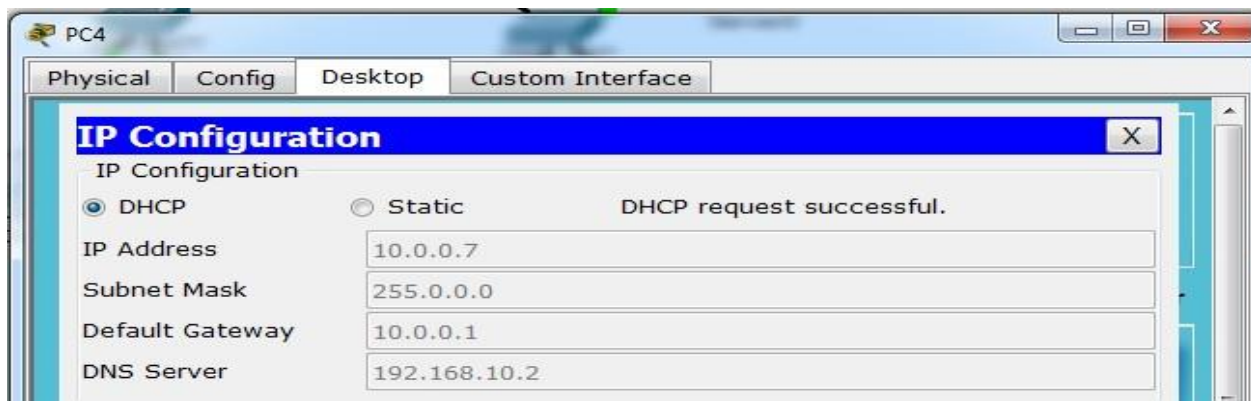
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#ip dhcp pool fast
Router(dhcp-config)#network 10.0.0.0 255.0.0.0
Router(dhcp-config)#default-router 10.0.0.1
Router(dhcp-config)#dns-server 192.168.10.2
Router(dhcp-config)#ip dhcp excluded-address 10.0.0.1 10.0.0.5
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#
```

## Now assigning IP to PC3 and PC4.





You can check the status of assigned IP addresses as shown below.

```
Router#show ip dhcp binding
```

IP address	Client-ID/ Hardware address	Lease expiration	Type
10.0.0.6	000A.F385.2E85	--	Automatic
10.0.0.7	0060.5C58.2C01	--	Automatic

### Now configuring network on Fa 0/1.

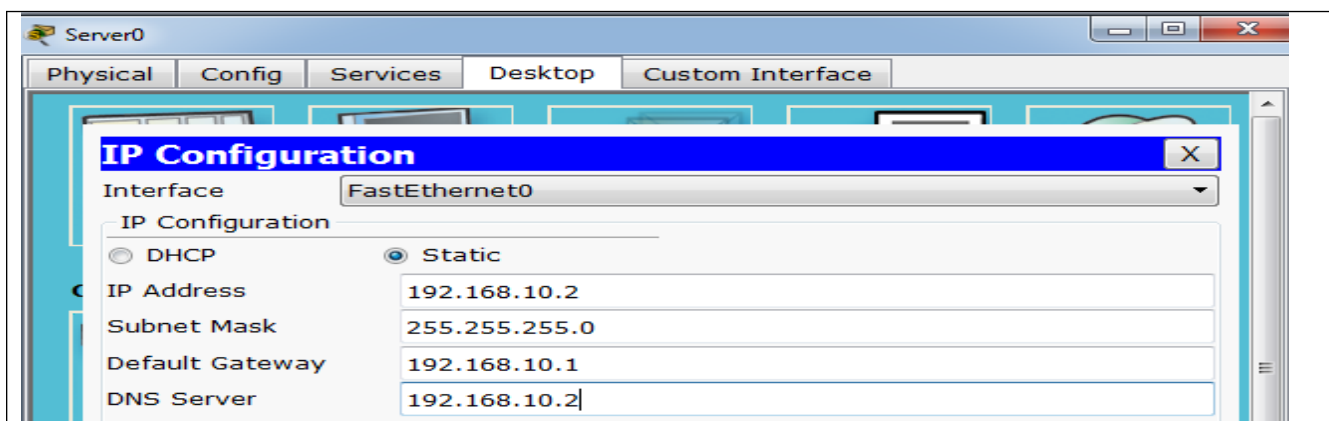
Click on router and assign IP address.

```
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/1
Router(config-if)#ip address 192.168.10.1 255.255.255.0
Router(config-if)#no shutdown

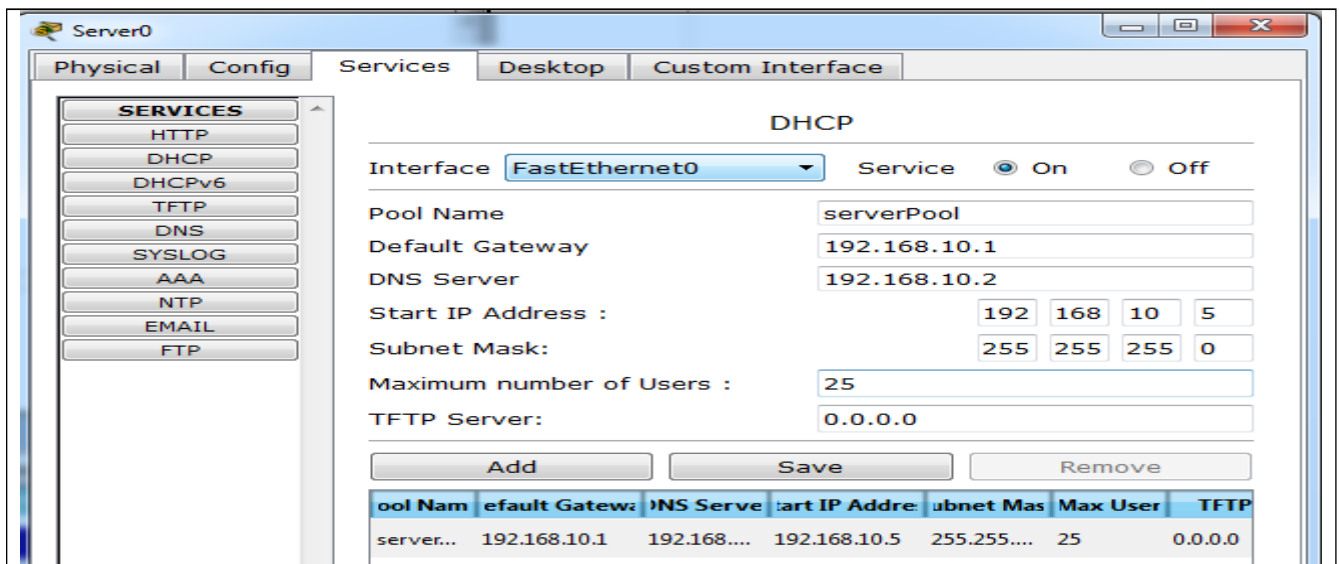
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
```

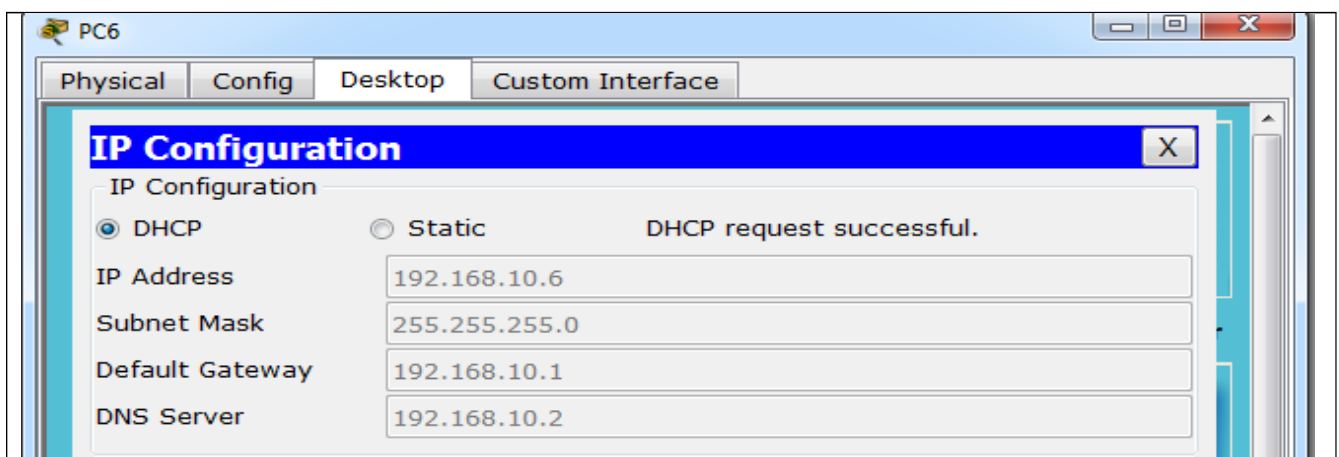
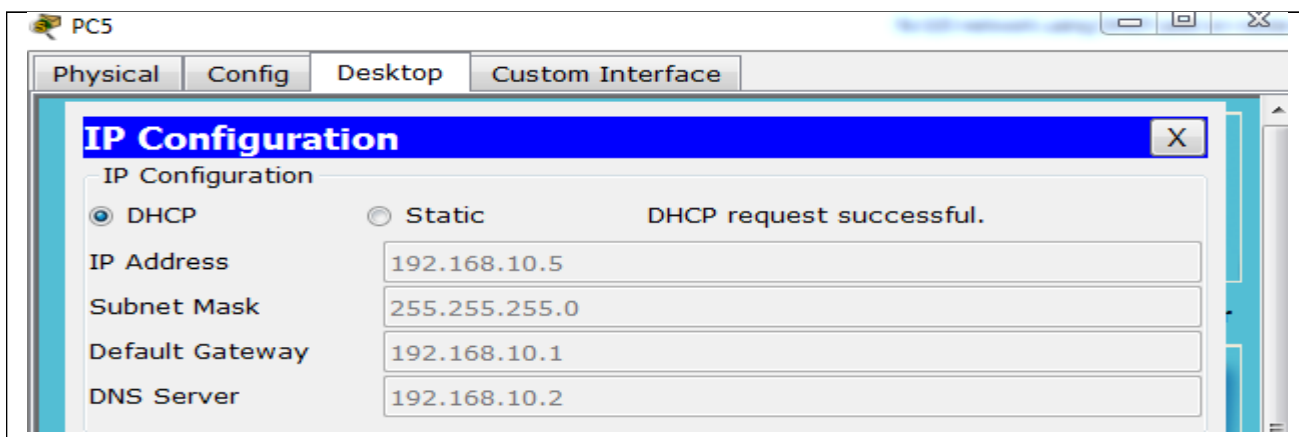
Click on server and assign IP address.



Now assigning DHCP pool on Server. Go to server →services →DHCP



Now assigning IP to PC5 and PC6.



### SIMULATION:

- Now click on simulation icon in the right bottom of packet Tracer.
- Now click on auto capture /play icon for packet capturing.
- Click on the PC and go to Desktop → IP configuration → DHCP

Event List					
Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.000	--	PC5	DHCP	
	0.000	--	PC5	DHCP	
	0.001	PC5	Switch4	DHCP	
	0.001	--	PC5	DHCP	
	0.002	PC5	Switch4	DHCP	
	0.002	Switch4	Router1	DHCP	
	0.002	Switch4	PC6	DHCP	
	0.002	Switch4	Server0	DHCP	
	0.003	Switch4	Router1	DHCP	

**Now click on the DHCP packet see how it lease IP address.**

### Requesting

OSI Model	Inbound PDU Details
At Device: Server0 Source: PC5 Destination: 255.255.255.255	
<b>In Layers</b>	<b>Out Layers</b>
Layer 7: DHCP Frame Server: 0.0.0.0, Client: 0.0.0.0	Layer7
Layer6	Layer6
Layer5	Layer5
Layer 4: UDP Src Port: 68, Dst Port: 67	Layer4
Layer 3: IP Header Src. IP: 192.168.10.5, Dest. IP: 255.255.255.255	Layer3
Layer 2: Ethernet II Header 0030.F217.9616 >> FFFF.FFFF.FFFF	Layer2
Layer 1: Port FastEthernet0	Layer1

0	8	16	31	Bits
OP: 0x1		HW TYPE		
TRANSACTION ID (4 BYTES)		HW LEN		
SECS		FLAGS		
CLIENT ADDRESS: 0.0.0.0				
"YOUR" CLIENT ADDRESS: 0.0.0.0				
SERVER ADDRESS: 0.0.0.0				
RELAY AGENT ADDRESS: 0.0.0.0				
CLIENT HARDWARE ADDRESS: 0030.F217.9616				
SERVER HOSTNAME (64 BYTES)				
FILE (128 BYTES)				
OPTIONS (312 BYTES)				

**Leased.**

### DHCP

0	8	16	31	Bits
OP: 0x2	HW TYPE	HW LEN	HOPS	
TRANSACTION ID (4 BYTES)				
SECS		FLAGS		
CLIENT ADDRESS: 0.0.0.0				
"YOUR" CLIENT ADDRESS: 192.168.10.7				
SERVER ADDRESS: 192.168.10.2				
RELAY AGENT ADDRESS: 0.0.0.0				
CLIENT HARDWARE ADDRESS: 0030.F217.9616				
SERVER HOSTNAME (64 BYTES)				
FILE (128 BYTES)				
OPTIONS (312 BYTES)				

a) **Shows OSI layers involved in transmission.**

The popped up window (below) will enable you to trace the content of the message through the OSI layer and what changes will occur at each layer (use next and previous buttons to trace each layer content).

PDU Information at Device: Server0

OSI Model   Inbound PDU Details   Outbound PDU Details

At Device: Server0  
Source: Server0  
Destination: Broadcast

**In Layers**

Layer 7: DHCP Frame Server: 192.168.10.2, Client: 0.0.0.0
Layer 6
Layer 5
Layer 4: UDP Src Port: 68, Dst Port: 67
Layer 3: IP Header Src. IP: 0.0.0.0, Dest. IP: 255.255.255.255
Layer 2: Ethernet II Header 0001.C77D.B427 >> FFFF.FFFF.FFFF
Layer 1: Port FastEthernet0

**Out Layers**

Layer 7: DHCP Frame Server: 192.168.10.2, Client: 0.0.0.0
Layer 6
Layer 5
Layer 4: UDP Src Port: 67, Dst Port: 68
Layer 3: IP Header Src. IP: 192.168.10.2, Dest. IP: 255.255.255.255
Layer 2: Ethernet II Header 0001.C786.AC87 >> FFFF.FFFF.FFFF
Layer 1: Port(s): FastEthernet0

# **Domain Name System**

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 various 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:**

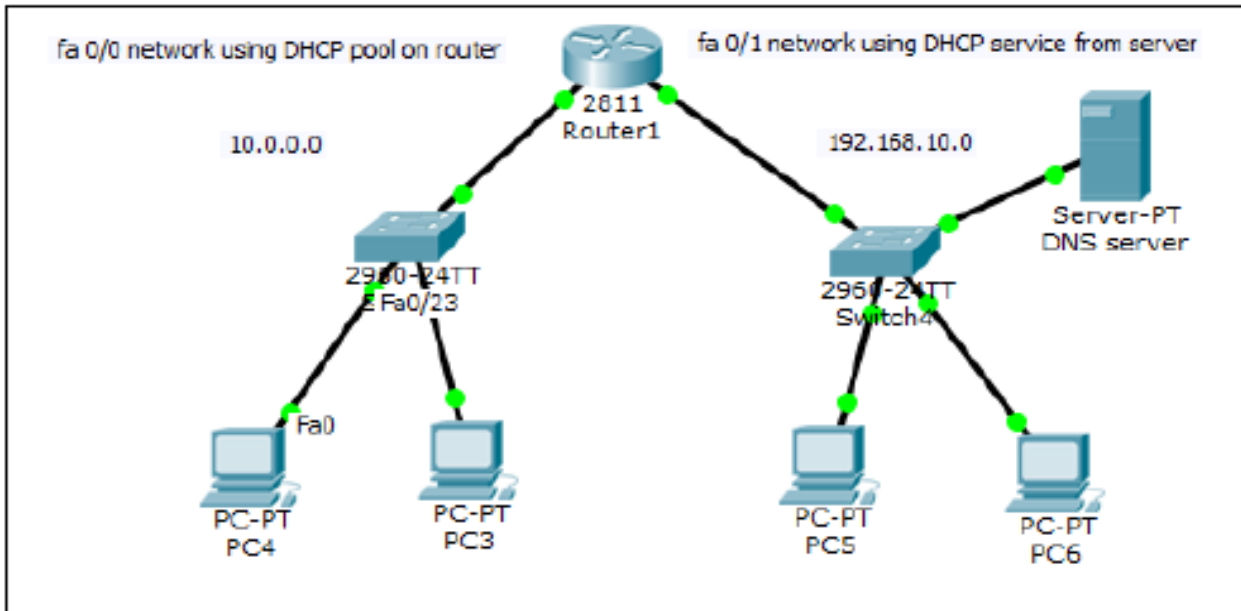
Canonical name records, or CNAME records, are often called alias records because they map an alias to the canonical name. When a name server finds a CNAME record, it replaces the name with the canonical name and looks up the new name. This allows pointing multiple systems to one IP without assigning 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.

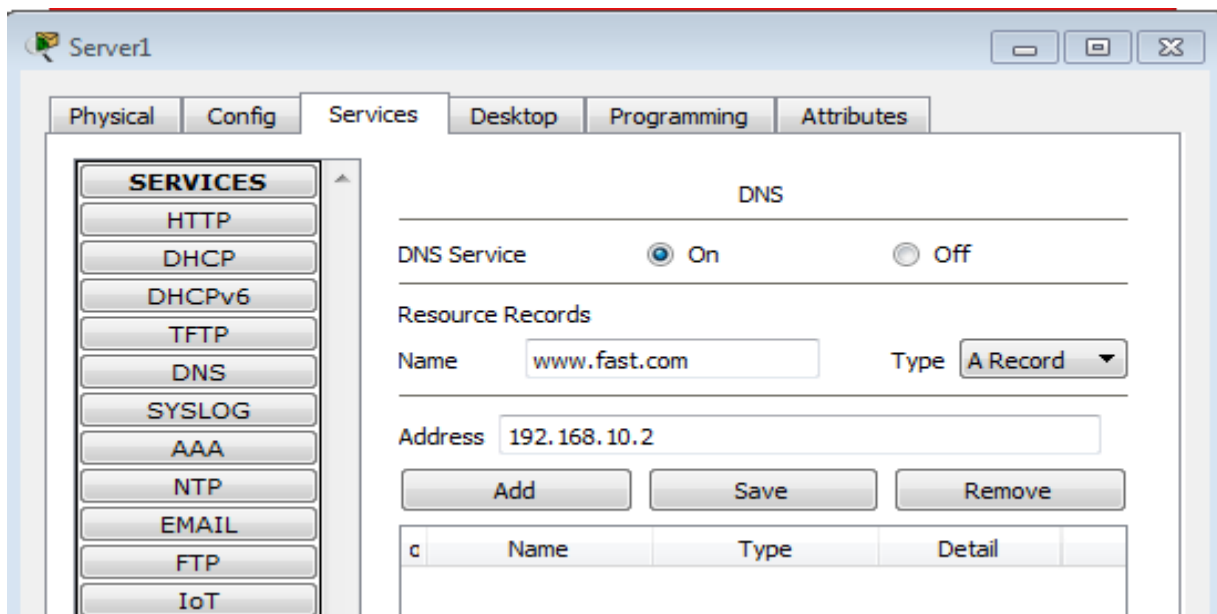
## SOA record:

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 using the DNS service on Server0. Go to server → services → DNS

First we add A record





Now click on Add.

The screenshot shows the 'Services' tab in a configuration window. On the left, a 'SERVICES' list includes HTTP, DHCP, DHCPv6, TFTP, DNS, SYSLOG, AAA, NTP, EMAIL, FTP, IoT, and VM Management. The 'DNS' service is selected. On the right, the 'DNS' configuration section shows 'DNS Service' set to 'On'. Under 'Resource Records', the 'Name' field is empty, 'Type' is set to 'A Record', and the 'Address' field is empty. Below these fields are 'Add', 'Save', and 'Remove' buttons. A table at the bottom shows a single record: '0 www.fast.com A Record 192.168.10.2'.

c	Name	Type	Detail
0	www.fast.com	A Record	192.168.10.2

Now add Cname record.

The screenshot shows the 'DNS' configuration section with 'DNS Service' set to 'On'. Under 'Resource Records', the 'Name' field is 'www.fast-cs.com', 'Type' is 'CNAME', and the 'Host Name' field is 'www.fast.com'. Below these fields are 'Add', 'Save', and 'Remove' buttons. A table at the bottom shows two records: '0 www.fast.com A Record 192.168.10.2' and '1 www.fast-cs.com CNAME www.fast.com'.

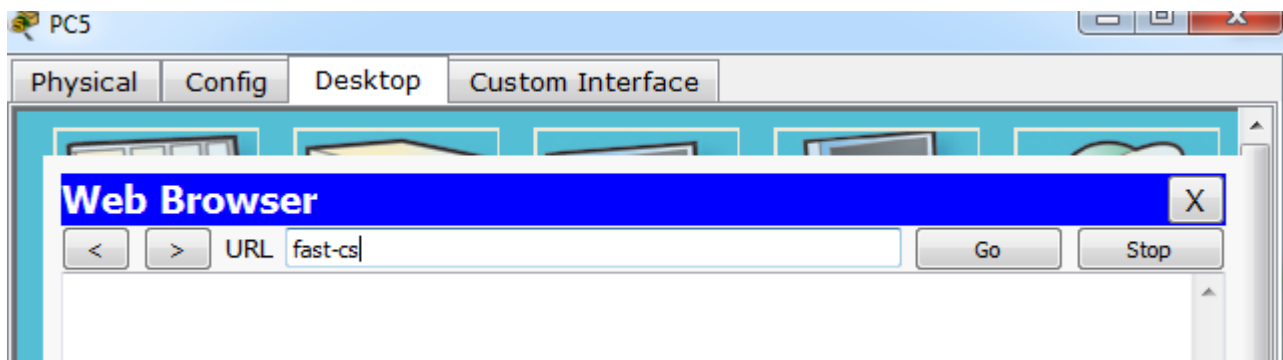
c	Name	Type	Detail
0	www.fast.com	A Record	192.168.10.2
1	www.fast-cs.com	CNAME	www.fast.com

Now click on Add.

The screenshot shows the 'DNS' configuration section with 'DNS Service' set to 'On'. Under 'Resource Records', the 'Name' field is empty, 'Type' is 'CNAME', and the 'Host Name' field is empty. Below these fields are 'Add', 'Save', and 'Remove' buttons. A table at the bottom shows three records: '0 www.fast-cs.com CNAME www.fast.com', '1 www.fast.com A Record 192.168.10.2', and a new record '2 www.fast-cs.com CNAME www.fast.com'.

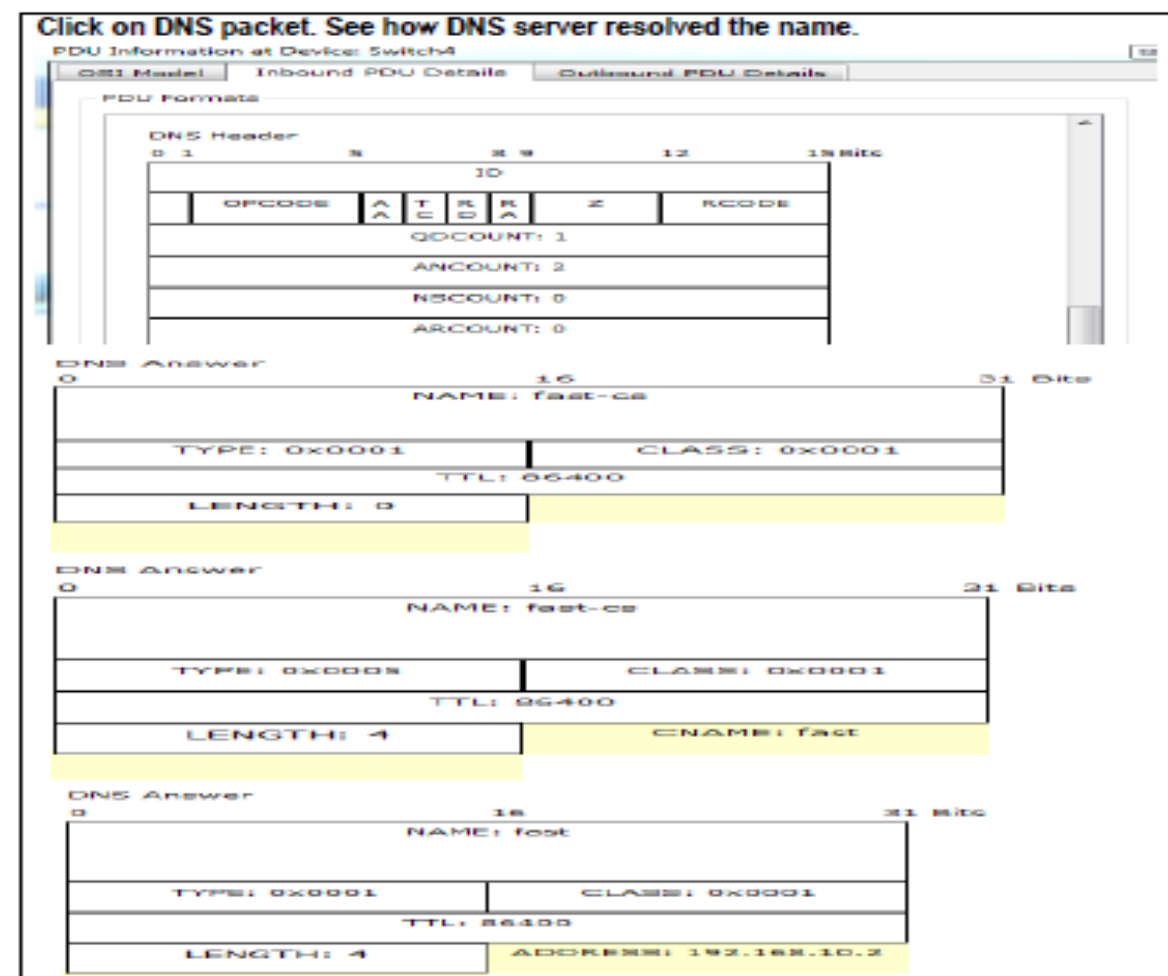
c	Name	Type	Detail
0	www.fast-cs.com	CNAME	www.fast.com
1	www.fast.com	A Record	192.168.10.2
2	www.fast-cs.com	CNAME	www.fast.com

Now go to p c5 → Desktop → web browser → type www. fast-cs.com and see how DNS works.



Start simulation.

Simulation Panel					
Event List					
Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.006	Switch4	DNS server	DNS	
	0.007	DNS server	Switch4	DNS	
	0.008	--	PC5	TCP	
	0.008	Switch4	PC5	DNS	
	0.008	--	PC5	TCP	
	0.009	PC5	Switch4	TCP	
	0.010	Switch4	DNS server	TCP	
	0.011	DNS server	Switch4	TCP	
	0.012	Switch4	PC5	TCP	

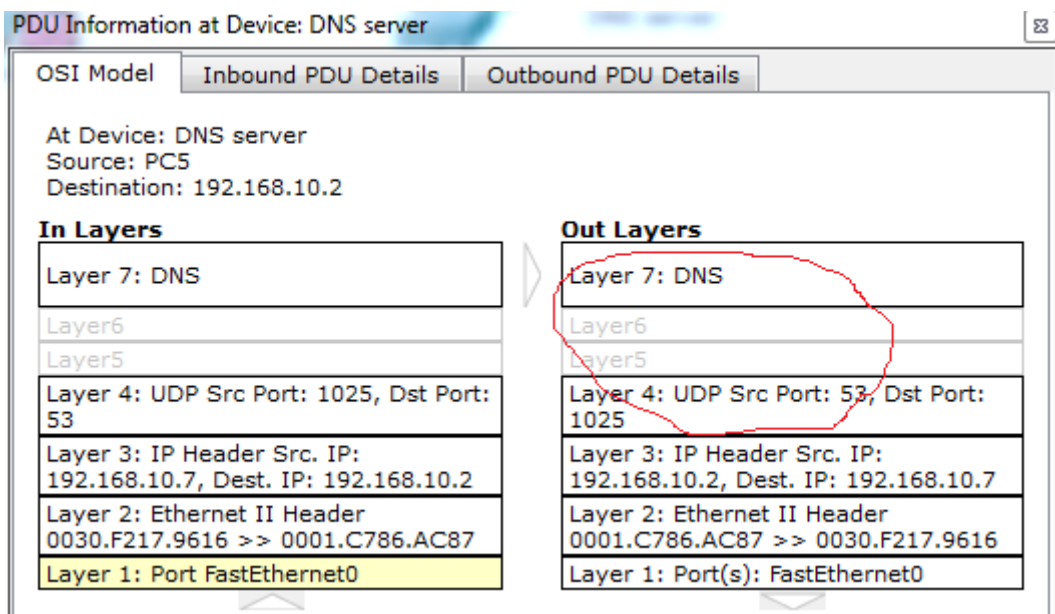


Now finally, required web page.

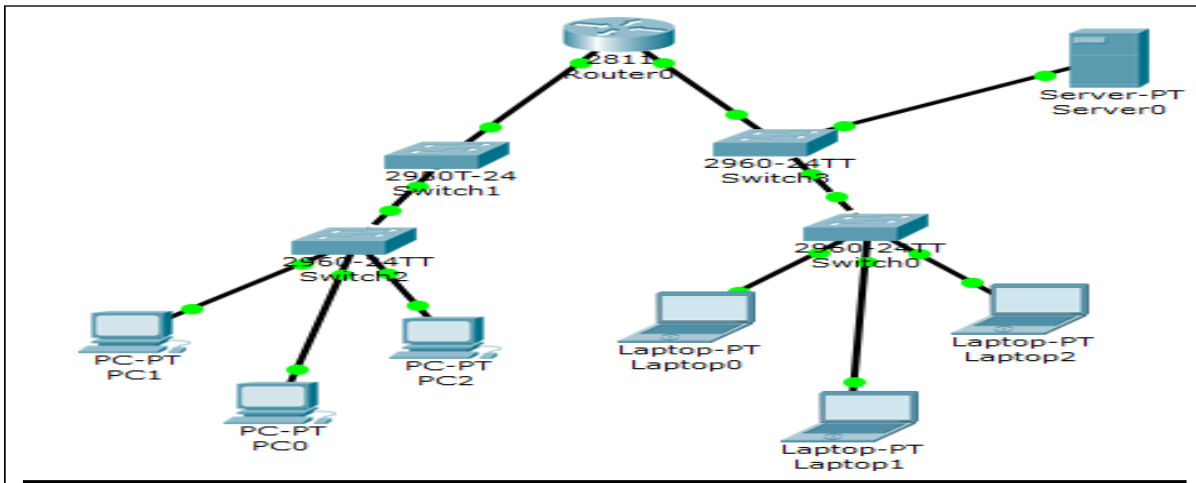


a) Shows OSI layers involved in transmission.

The popped up window (below) will enable you to trace the content of the message through the OSI layer and what changes will occur at each layer (use next and previous buttons to trace each layer content).



## **EXERCISE:**



**Try to implement the topology this topology using DHCP service from router and DNS from server.**

Answer the following questions:

1. Are DHCP messages sent over UDP or TCP?
2. What is the link-layer (e.g., Ethernet) address of your host?
3. What values in the DHCP discover message differentiate this message from the DHCP request message?
4. What is the value of the Transaction-ID in each of the first four (Discover/Offer/Request/ACK) DHCP messages? What are the values of the Transaction-ID in the second set (Request/ACK) set of DHCP messages? What is the purpose of the Transaction-ID field?
5. A host uses DHCP to obtain an IP address, among other things. But a host's IP address is not confirmed until the end of the four-message exchange! If the IP address is not set until the end of the four-message exchange, then what values are used in the IP datagrams in the four-message exchange? For each of the four DHCP messages (Discover/Offer/Request/ACK DHCP), indicate the source and destination IP addresses that are carried in the encapsulating IP datagram.
6. What is the IP address of your DHCP server?
7. What IP address is the DHCP server offering to your host in the DHCP Offer message? Indicate which DHCP message contains the offered DHCP address.
8. In the example screenshot in this assignment, there is no relay agent between the host and the DHCP server. What values in the trace indicate the absence of a relay agent? Is there a relay agent in your experiment? If so what is the IP address of the agent?
9. Explain the purpose of the lease time. How long is the lease time in your experiment?

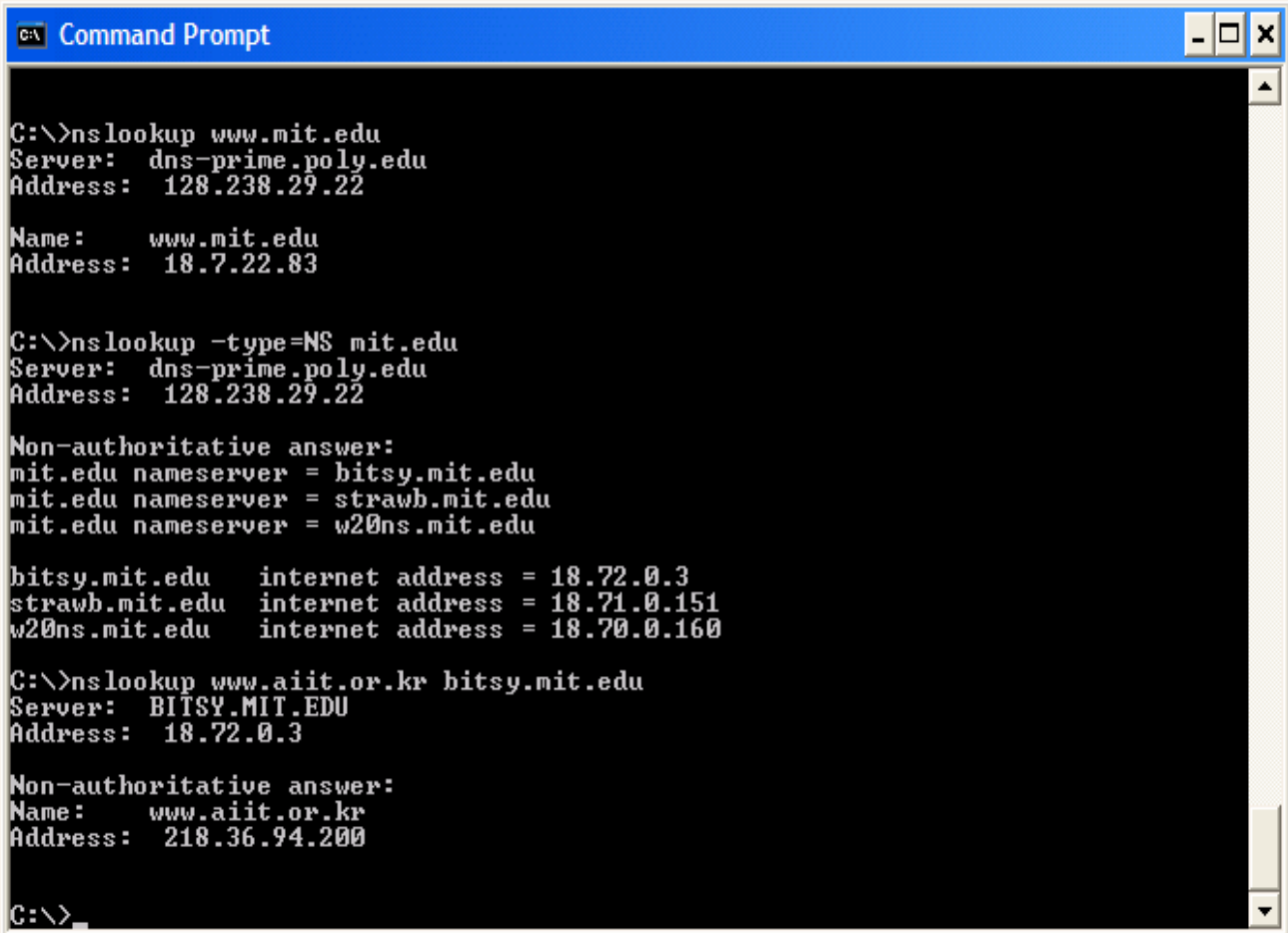
## DNS EXPERIMENT:

The Domain Name System (DNS) translates hostnames to IP addresses, fulfilling a critical role in the Internet infrastructure. In this lab, we'll take a closer look at the client side of DNS. Recall that the client's role in the DNS is relatively simple – a client sends a *query* to its local DNS server, and receives a *response* back. From the DNS client's standpoint, however, the protocol is quite simple – a query is formulated to the local DNS server and a response is received from that server.

### 1. nslookup

In this lab, we'll make extensive use of the *nslookup* tool, which is available in most Linux/Unix and Microsoft platforms today. To run *nslookup* in Linux/Unix, you just type the *nslookup* command on the command line. To run it in Windows, open the Command Prompt and run *nslookup* on the command line.

In its most basic operation, *nslookup* tool allows the host running the tool to query any specified DNS server for a DNS record. The queried DNS server can be a root DNS server, a top-level-domain DNS server, an authoritative DNS server, or an intermediate DNS server (see the textbook for definitions of these terms). To accomplish this task, *nslookup* sends a DNS query to the specified DNS server, receives a DNS reply from that same DNS server, and displays the result.



```
C:\>nslookup www.mit.edu
Server:  dns-prime.poly.edu
Address:  128.238.29.22

Name:    www.mit.edu
Address:  18.7.22.83

C:\>nslookup -type=NS mit.edu
Server:  dns-prime.poly.edu
Address:  128.238.29.22

Non-authoritative answer:
mit.edu nameserver = bitsy.mit.edu
mit.edu nameserver = strawb.mit.edu
mit.edu nameserver = w20ns.mit.edu

bitsy.mit.edu    internet address = 18.72.0.3
strawb.mit.edu   internet address = 18.71.0.151
w20ns.mit.edu    internet address = 18.70.0.160

C:\>nslookup www.aiit.or.kr bitsy.mit.edu
Server:  BITSY.MIT.EDU
Address:  18.72.0.3

Non-authoritative answer:
Name:    www.aiit.or.kr
Address:  218.36.94.200

C:\>
```

The above screenshot shows the results of three independent *nslookup* commands (displayed in the Windows Command Prompt). In this example, the client host is located on the campus of Polytechnic University in Brooklyn, where the default local DNS server is dns-prime.poly.edu. When running *nslookup*, if no DNS server is specified, then *nslookup* sends the query to the default DNS server, which in this case is dns-prime.poly.edu.

Consider the first command: `Nslookup www.mit.edu`

In words, this command is saying “please send me the IP address for the host `www.mit.edu`”. As shown in the screenshot, the response from this command provides two pieces of information: (1) the name and IP address of the DNS server that provides the answer; and (2) the answer itself, which is the host name and IP address of `www.mit.edu`. Although the response came from the local DNS server at Polytechnic University, it is quite possible that this local DNS server iteratively contacted several other DNS servers to get the answer, as described in Section 2.4 of the textbook.

Now consider the second command: `nslookup -type=NS mit.edu`

In this example, we have provided the option “-type=NS” and the domain “mit.edu”. This causes *nslookup* to send a query for a type-NS record to the default local DNS server. In words, the query is saying, “please send me the host names of the authoritative DNS for mit.edu”. (When the -type option is not used, *nslookup* uses the default, which is to query for type A records.) The answer, displayed in the above screenshot, first indicates the DNS server that is providing the answer (which is the default local DNS server) along with three MIT name servers. Each of these servers is indeed an authoritative DNS server for the hosts on the MIT campus. However, *nslookup* also indicates that the answer is “non-authoritative,” meaning that this answer came from the cache of some server rather than from an authoritative MIT DNS server. Finally, the answer also includes the IP addresses of the authoritative DNS servers at MIT. (Even though the type-NS query generated by *nslookup* did not explicitly ask for the IP addresses, the local DNS server returned these “for free” and *nslookup* displays the result.)

Now finally consider the third command: `nslookup www.aiit.or.kr bitsy.mit.edu`.

In this example, we indicate that we want the query sent to the DNS server `bitsy.mit.edu` rather than to the default DNS server (`dns-prime.poly.edu`). Thus, the query and reply transaction takes place directly between our querying host and `bitsy.mit.edu`. In this example, the DNS server `bitsy.mit.edu` provides the IP address of the host `www.aiit.or.kr`, which is a web server at the Advanced Institute of Information Technology (in Korea).

Now that we have gone through a few illustrative examples, you are perhaps wondering about the general syntax of *nslookup* commands.

The syntax is: `nslookup -option1 -option2 host-to-find dns-server`

In general, *nslookup* can be run with zero, one, two or more options. And as we have seen in the above examples, the `dns-server` is optional as well; if it is not supplied, the query is sent to the default local DNS server.

Now that we have provided an overview of *nslookup*, it is time for you to test drive it yourself. Do the following (and write down the results):

Now that we have provided an overview of *nslookup*, it is time for you to test drive it yourself. Do the following (and write down the results):

1. Run *nslookup* to obtain the IP address of a Web server in Asia. What is the IP address of that server?
2. Run *nslookup* to determine the authoritative DNS servers for a university in Europe.
3. Run *nslookup* so that one of the DNS servers obtained in Question 2 is queried for the mail servers for Yahoo! mail. What is its IP address?

Now that we are familiar with *nslookup* and *ipconfig*, we’re ready to get down to some serious business. Let’s first capture the DNS packets that are generated by ordinary Web-surfing activity.

- Use ipconfig to empty the DNS cache in your host.
- Open your browser and empty your browser cache. (With Internet Explorer, go to Tools menu and select Internet Options; then in the General tab select Delete Files.)
- Open Wireshark and enter “ip.addr == your\_IP\_address” into the filter, where you obtain your\_IP\_address with ipconfig. This filter removes all packets that neither originate nor are destined to your host.
- Start packet capture in Wireshark.
- With your browser, visit the Web page: <http://www.ietf.org>
- Stop packet capture.

If you are unable to run Wireshark on a live network connection, you can download a packet trace file that was captured while following the steps above on one of the author's computers<sup>1</sup>. Answer the following questions. Whenever possible, when answering a question below, you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. Annotate the printout<sup>2</sup> to explain your answer. To print a packet, use File->Print, choose selected packet only, choose Packet summary line, and select the minimum amount of packet detail that you need to answer the question.

1. Locate the DNS query and response messages. Are they sent over UDP or TCP?
2. What is the destination port for the DNS query message? What is the source port of the DNS response message?
3. To what IP address is the DNS query message sent? Use ipconfig to determine the IP address of your local DNS server. Are these two IP addresses the same?
4. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
5. Examine the DNS response message. How many “answers” are provided? What do each of these answers contain?