Lab

6

Network Address Translation (NAT)

Dynamic Host Configuration Protocol (DHCP)

What you will learn in this lab:

- How NAT (Network Address Translation) works.
- How DHCP (Dynamic Host Configuration Protocol) works.
- How DHCP works together with NAT.

Updated: November 2020

Table of Content

Study Material for Lab 6

1. **Unix commands for NAT, DHCP:** Go to the Linux manual pages at
   http://man7.org/linux/man-pages/

   and read the manual pages of the following commands:
   - `iptables`
   - `dhcpcd`
   - `dhcpd`
   - `dhcpd.conf`
   - `dhcp-options`
   - `dhcpd.leases`

2. **Private IP addresses:** Read RFC 1918 on address allocation in private networks
   https://tools.ietf.org/html/rfc1918

3. **Network Address Translation (NAT):** Read the following tutorial on NAT at
   https://www.softwaretestinghelp.com/network-address-translation-nat/

4. **Netfilter/iptables** Read about `netfilter` and `iptables` at
   https://www.karlrupp.net/en/computer/nat_tutorial

5. **Dynamic Host Configuration Protocol (DHCP):** Read about DHCP at
   https://www.computernetworkingnotes.com/ccna-study-guide/how-dhcp-works-explained-with-examples.html

   https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol

Question Sheet for Prelab 6

1. Explain why NAT is often mentioned as a solution to counteract the depletion of IPv4 addresses? Which alternatives to NAT exist that address the scarcity of available IPv4 addresses?
2. What does the following comment refer to: “NAT destroys the ability to do host-to-host communication over the Internet”?
3. Explain the following terms which are used in the context of Network Address Translation:
   a. Static NAT
   b. Dynamic NAT
   c. NAT with IP overload
   d. Port Address Translations
   e. IP Masquerading
4. Refer to RFC 1918 and list the IP address blocks that are reserved for use in private networks. Why is there a need to specify IP addresses for private networks?
5. The utility *netfilter* and the command *iptables* provide support for NAT in Linux systems. Explain the relationship between the *netfilter* utility and the *iptables* command?
6. Describe the following terms which are used in the *iptables* command:
   a. Chain
   b. Postrouting
   c. Prerouting
7. Consider a NAT device between a private and the public network. Suppose the private network uses addresses in the range 10.0.1.0-10.0.1.255, and suppose that the interface of the NAT device to the public network has IP address 128.143.136.80.
   a. Write the iptables command so that the addresses in the private network are mapped to the public IP address 128.143.136.80.
   b. Write an IOS command so that the addresses in the private network are mapped to the public IP address 128.143.136.80.
8. Explain the meaning of the “magic cookie” in the DHCP protocol.
9. If the command *dhcpd* is issued (without arguments) on a Linux PC with multiple network interfaces, which network interfaces does the DHCP server listen on?

Lab 6

Figure 1 shows two private networks, both 10.0.1.0/24, which are connected to a public network. Each private network is connected to the public network by a NAT device, which is either a PC or a Cisco router. On each NAT device, IP forwarding must be enabled.

(Note: In the private networks in Figure 1, Router1 and Router3 are used to mimic hosts, i.e., they are not configured to act as IP routers.)

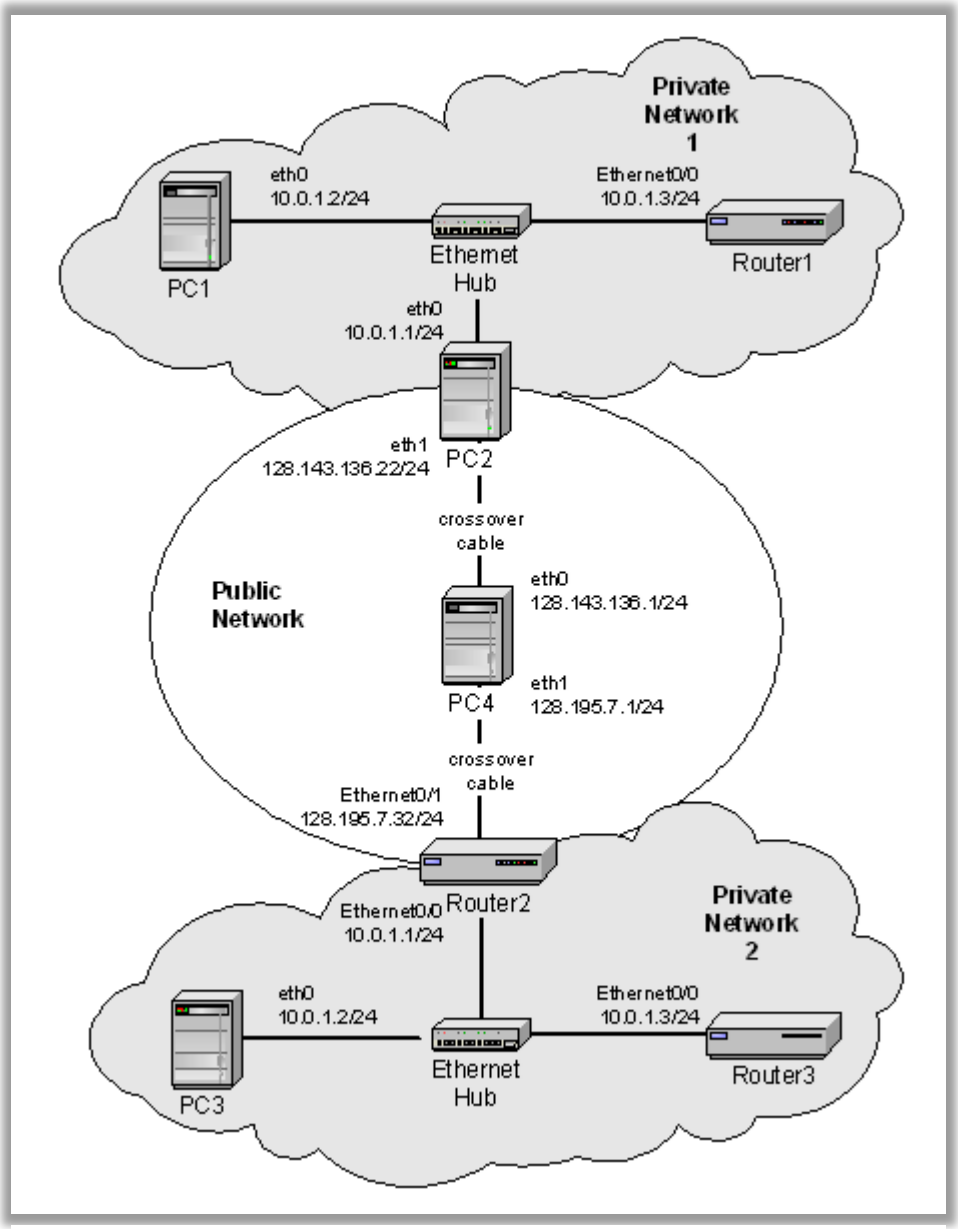


Figure 1. Configuration for Part 1.

- In this lab, PC2 and Router2 are IPv4 routers that provide the interconnection of the private networks to the public network. Both PC2 and Router2 are configured as NAT devices.
- On PC2, the *Netfilter* utility provides the ability to set IP packet filters, including NAT functions. On Router2, you will use Cisco IOS commands to configure NAT rules.
- PC4 runs as an IPv4 router.
- The assignment of IPv4 addresses and default gateways the PCs and routers are shown in Table1 and Table 2.

Table 1. IPv4 addresses and gateways assignment of the Linux PCs for Part 1.

Machine	IP Address of Internet Interface (eth0)	IP Address of Private Network Interface (eth1)	Default Gateway
PC1	10.0.1.2/24	none	10.0.1.1
PC2	10.0.1.1/24	128.143.136.22/24	128.143.136.1
PC3	10.0.1.2/24	none	10.0.1.1
PC4	128.143.136.1/24	128.195.7.1/24	none

Table 2. IPv4 addresses and gateway assignment of the Cisco routers for Part 1.

Machine	IP Address of Interface (Ethernet0/0)	IP Address of Interface (Ethernet0/1)	Default Gateway
Router1	10.0.1.3/24	None	10.0.1.1
Router2	10.0.1.1/24	128.195.7.32/24	128.195.7.1
Router3	10.0.1.3/24	None	10.0.1.1

Part 1. NAT (Network Address Translation)

NAT (Network Address Translation) refers to a function that replaces the IP addresses (and possibly the port numbers) of IP datagrams. NAT is run on routers that connect private networks to the public Internet, to replace the IP address-port pair of an IP packet with another IP address-port pair. Generally, the operations of NAT are specified in terms of a set of rules which determines how IP addresses are to be replaced.

Often, a NAT device is referred to as a NAT box. One of the reasons for using NAT is that it conserves IP addresses. NAT allows hosts in a private network to share public IP addresses, or to limit the use of public IP addresses to a small number of hosts in the private network.

Private networks may have IP addresses that are non-Internet routable, as specified in RFC 1918. This means that the Internet routers do not have entries in their routing tables for these addresses.

In the network in Figure 1, both PC2 and Router2 will be configured as NAT devices. With NAT, the hosts in the private networks can access the public network, i.e., they are able to reach the addresses on the 128.143.136.0/24 and 128.195.7.0/24 networks.

Exercise 1.a- Network Setup

Configure the network in Figure 1 with the IP address configuration shown in **Error! Reference source not found.** and **Error! Reference source not found.**. The following commands review the steps involved in the configuration.

Step 1: On the Linux PCs, configure the IP addresses of the interfaces and add a default gateway as shown in Table 1. The commands for PC1 are

```
PC1$ sudo ip addr add 10.0.1.2/24 dev eth0
PC1$ sudo ip route add default via 10.0.1.1
```

Step 2: Enable IPv4 forwarding on PC2 and PC4. This is done with the command

```
PC2$ sudo sysctl -w net.ipv4.ip_forward=1
```

Step 3: Configure the IP addresses and default gateway of Router1 and Router3 as shown in Table 2. Both Router1 and Router3 are configured as hosts and therefore have IP forwarding disabled with the command `no ip routing'. The commands for the configuration of Router1 are

```
Router1# configure terminal
Router1(config)# no ip routing
Router1(config)# ip default-gateway 10.0.1.1
Router1(config)# interface Ethernet0/0
Router1(config-if)# ip address 10.0.1.3 255.255.255.0
Router1(config-if)# no shutdown
Router1(config-if)# end
```

ip default-gateway vs. ip route 0.0.0.0:
Cisco IOS has different commands for setting the default route on a host (no ip routing) and on an IP router (ip routing). On Router1, which is configured as a host, the command was
`Router1(config)# ip default-gateway 10.0.1.1`

If Router1 had been configured as an IP router, the command would be
`Router1(config)# ip route 0.0.0.0 0.0.0.0 10.0.1.1`

Even though there is no conceptual difference, Cisco IOS uses different terms to refer the next hop router, e.g., when displaying the routing table. On a host, it is called *default gateway*, and on an IP router, it is *gateway of last resort*.

Step 4: Configure the IPv4 addresses and default route of Router2 as shown in **Error! Reference source not found.**.

```
Router2# configure terminal
Router2(config)# no ip routing
Router2(config)# ip routing
Router2(config)# ip route 0.0.0.0 0.0.0.0 128.195.7.1
Router2(config)# interface Ethernet0/0
Router2(config-if)# ip address 10.0.1.1 255.255.255.0
Router2(config-if)# no shutdown
Router2(config-if)# exit
Router2(config)# interface Ethernet0/1
Router2(config-if)# ip address 128.195.7.32 255.255.255.0
Router2(config-if)# no shutdown
Router2(config-if)# end
```

In the above commands, the command `ip route 0.0.0.0 0.0.0.0 128.195.7.1` sets the default gateway for Router2.

After completing the setup of the configuration, you should be able to issue successful `ping` commands between hosts in the private network, and between hosts in the public network. However, `ping` commands across a private/public network boundary are not successful. Verify this!

Exercise 1.b- Configuration of NAT on a Cisco Router

Next, you configure a Cisco router as a NAT device.

Some Cisco Conventions
In Cisco IOS, the private network is referred to as "*inside*", and the public network is referred to as "*outside*". An IP address that is seen by hosts on the inside is called a local address, and an IP address that is seen by hosts on the outside is called a global address. There are four different types of addresses:

- An ***inside local*** address is an address in the private network that is not visible in the public network.
- An ***inside global*** address can be used in the public network for devices in the private network.
- An ***outside global*** address is an address in the public network.
- An ***outside local*** address is used by devices in the private network to refer to an address in the public network.

Using this terminology, a NAT device translates ***inside local*** addresses to ***outside global*** addresses and ***outside global*** addresses to ***inside local*** addresses.

Step 1: **Modify the NAT table of Router2:** Use the following commands to set up Router2 as a NAT device.

a. The NAT translation table is displayed with the command
`Router2# show ip nat translations`

b. Add a NAT rule so that the private IP address of PC3, 10.0.1.2, is translated to the public address 200.0.0.2.

Table 3. Private and public address of PC3.

Inside local address	Outside global address
10.0.1.2/24	200.0.0.2/24

The IOS commands for this are as follows

```
Router2# configure terminal
Router2(config)# interface Ethernet0/0
Router2(config-if)# ip nat inside
Router2(config-if)# exit
Router2(config)# interface Ethernet0/1
Router2(config-if)# ip nat outside
Router2(config-if)# exit
Router2(config)# ip nat inside source static 10.0.1.2 200.0.0.2
Router2(config)# end
```

c. After the above rule has been entered, display the content of the NAT table again with
`Router2# show ip nat translations`

Take a screenshot of the NAT table.

The commands which are used above are explained as follows

IOS mode: Privileged EXEC
`show ip nat translations`
Displays the content of the NAT table.

IOS mode: Interface configuration

`ip nat inside`

Specifies that an interface is connected to the private network.

`ip nat outside`

Specifies that an interface is connected to the public network.

IOS Mode: Global Configuration

`ip nat inside source static <IPaddr1> <IPaddr2>`

Adds a rule so that the private IP address IPaddr1 is mapped to the public IP address IPaddr2

For example, the command,

```
ip nat inside source static 10.0.1.2 200.0.0.2
```

Adds a rule so that the private address 10.0.1.2 is mapped to the public address 200.0.0.2

Dynamic NAT table entries:
"Dynamic NAT" is an alternative to the static NAT table entries used in this exercise. With dynamic NAT, a pool of global addresses is specified at the NAT device. Addresses from the pool are dynamically mapped to the private addresses whenever there is a demand for a new address.

Step 2: **Update routing tables:** Add a static routing entry to the routing table of PC4, so that traffic with destination IP address `200.0.0.0/24` is forwarded to Router2.

Step 3: **Observe traffic at a NAT device:** To observe the IP address translation, issue `ping` commands between machines in the public and private network. Use *Wireshark* to capture packets on the private and public interfaces of Router2.

- Start a *Wireshark* session on PC3 (eth0) to capture the traffic from Router2 on the private network.
- Start a *Wireshark* session on interface eth1 of PC4 to capture the traffic from Router2 on the public network.
- Issue the following ping commands:
- On PC3:

```
PC3$ ping -c3 10.0.1.3
PC3$ ping -c3 128.143.136.1
```

On Router3:

```
Router3# ping 10.0.1.2
Router3# ping 128.143.136.1
```

On PC4:

```
PC4$ ping -c3 10.0.1.2
PC4$ ping -c3 200.0.0.2
```

- Observe which `ping` commands succeed and take note of the results.

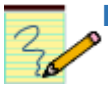
- Capture the IP source address and IP destination address from the IP header data of an ICMP request and the corresponding ICMP reply packet before and after it passes through Router2. You can do this by taking screenshots from the packet list pane of Wireshark.

Step 4: **Add additional NAT table entries:** Add NAT rules to Router2, so that Router2 and Router3 (on interface Etherenet0/0) are addressable from the public network. The private and public addresses are given in Table 4. Private and public addresses of Router2 and Router3.

Machine	Inside local address	Outside global address
Router2	10.0.1.1	200.0.0.1
Router3	10.0.1.3	200.0.0.3

Table 4. Private and public addresses of Router2 and Router3.

Issue the `ping` commands from Router3 and PC4 as specified above. Determine which ping command(s), which previously failed, is (are) now successful.



Lab Question/Report:

1. Include the screenshot of the NAT table of Router2 from Step 1. Explain the meaning of all columns in the table.
2. For each of the ping commands in Step 3, provide an explanation why the command succeeds or fails.
3. Include the screenshots showing the IP source address and IP destination address from the IP header data of an ICMP request and the corresponding ICMP reply packet before and after it passes through Router2.
4. Provide the Cisco IOS commands to add the NAT table entries in Step 4.
5. Determine which ping command(s), which failed in Step 4, was (were) successful in Step 4. Explain your answer.

Exercise 1.c- IP Masquerading with a Linux PC

In this exercise, we consider a special use of NAT that allows multiple private IP addresses to be mapped to a single public IP address. This use of NAT is called *IP masquerading*, *port address translation* (PAT) or

Network Address and Port Translation (NAPT). Here, the private network has only a single public IP address, but has multiple hosts in the private network. IP Masquerading modifies the port number of packets so that the single public IP address can be overloaded. This is the way almost all home routers function.

In this exercise, PC2 is configured to perform IP masquerading. This is done with the *Netfilter* utility, which adds the ability to set IP packet filters in a Linux system. IP packet filters are used to add firewalls as well as NAT functionality to a system. The `iptables` command is used to set up, maintain, and inspect IP packet filter rules to a Linux kernel.

PREROUTING and **POSTROUTING:**
On a Linux system, the configuration of NAT manipulates a set of rules of the Netfilter utility, called NAT table. The rules in the NAT table are grouped in so-called chains. Two of the built-in chains are called **PREROUTING** and **POSTROUTING**:

- **PREROUTING** – The rules in this chain are applied to incoming datagrams.
- **POSTROUTING** – The rules in this chain are applied to outgoing datagrams. The main rule is SNAT (Source Network Address Translation), which specifies how the source address of an outgoing IP datagram should be modified.

Commands that manipulate the NAT table start with

```
PC2% sudo iptables –t nat
```

The following are some of the most important commands that manipulate the NAT table.

Some Linux commands to manipulate the NAT table

`sudo iptables –t nat -L`
Displays/lists all rules in the NAT table.

`sudo iptables –t nat –D POSTROUTING 1`
Deletes the first rule in the POSTROUTING chain of the NAT table

`sudo iptables –t nat –F`
Deletes all entries (flushes) in the NAT table

`sudo iptables -t nat -A POSTROUTING -j SNAT -s` ***`IPAddr2/netmask`*** **`--to`** ***`IPAddr1`***
Adds the following rule to the POSTROUTING chain of the NAT table:
”in IP datagrams that go to the public network, the IP source address ***`IPAddr2/netmask`*** is changed to ***`IPAddr1`***”.

For example, in the following command

```
sudo iptables –t nat –A POSTROUTING –j SNAT --to 128.195.7.32 –s 10.0.1.0/24
```

the source address of outgoing IP datagrams that match 10.0.1.0/24 is changed to 128.195.7.32.

Step 1: **Modify the NAT table of PC2:** On PC2, add a rule to the NAT table so that the IP source address of all outgoing IP datagrams are set to IP address 128.143.136.22.
Display the content of the NAT table and take a screen snapshot.

Step 2: **Observe traffic at a NAT device:**

a. To observe the IP address translation, capture packets on both interfaces of PC2 that are between the private networks and the Internet. On PC2, run *Wireshark* on both *eth0* and *eth1*.

b. Start a Telnet server on PC4 and PC1. The command on PC1 is

```
PC1$ sudo service xinetd start
```

Telnet
Telnet is a remote terminal program that operates over a TCP connection. Since Telnet does not encrypt the payload, it is not widely used anymore, and has largely been replaced by ssh. TCP servers bind to the well-known TCP port 23.

Telnet performs a login on the remote system. On the Linux PCs, the username and password are both `*labuser*'. After the login, you can enter commands on the remote system.

To terminate a Telnet session, type `exit` or `Ctrl-d`.

c. Establish a set of Telnet session and login to remote machines, as given below. Once connected, terminate the Telnet session.

On PC1:

```
PC1$ telnet 128.143.136.1
```

On Router1:

```
Router1# telnet 10.0.1.2
Router1# telnet 128.143.136.1
```

On PC4:

```
PC4$ telnet 10.0.1.2
```

For each `telnet` command, provide an explanation why the command succeeds or fails.

For **each** successful Telnet session that traverses the NAT device (PC2), observe how the IP addresses and port numbers are mapped by PC2.

For **one** successful Telnet session that traverses the NAT device (PC2),

- Take screen captures that show the IP header and TCP port numbers of a packet that is sent from the private network to the public network. Capture the same packet twice: (1) before and (2) after it traverses PC2.
- Take screen captures that show the IP header and TCP port numbers of a packet that is sent from the public network to the private network. Capture the same packet twice: (1) before and (2) after it traverses PC2.

Step 3: **Observe mapping of ICMP packets:** The ping command sends out ICMP Echo Request messages and receives ICMP Echo Reply messages. Since ICMP messages do not contain a port number, it is not entirely obvious how a NAT device that performs IP masquerading can direct ICMP Echo Reply messages that return from the public network to the *private* network. In this exercise, you will explore how a NAT device handles ICMP messages.

a. On PC2, run *Wireshark* on both *eth0* and *eth1*. Use the appropriate filters to capture the traffic generated by *ping* commands.

b. Issue the following *ping* commands:

On PC1:

```
PC1$ ping -c3 10.0.1.3
PC1$ ping -c3 128.143.136.1
PC1$ ping -c3 200.0.0.3
```

On Router1:

```
Router1# ping 10.0.1.2
Router1# ping 128.143.136.1
Router1# ping 200.0.0.3
```

On PC4:

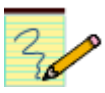
```
PC4$ ping -c3 10.0.1.2
```

For each `ping` command, provide an explanation why the command succeeds or fails.

c. Take screenshots of the output of the commands.

d. Save the *Wireshark* output and the output of ping commands into files.

Lab Question/Report:



1. Provide the screenshot of the NAT table of PC2 from Step 1.

2. For each successful Telnet session in Step 2 that traverses the NAT device (PC2), create a table that shows the IPv4 addresses and port numbers that are mapped by PC2.
3. Include the screen captures from Step 2 and label each screenshot.
4. Include the screen captures from Step 3 showing the output of the ping commands.
5. For each `ping` command in Step 3, provide an explanation why the command succeeds or fails.
6. When PC2 receives an ICMP Echo Reply from the public network, how does it determine whether to forward it to PC1 or Router?

Part 2. Dynamic Host Configuration Protocol (DHCP)

The Dynamic Host Configuration Protocol (DHCP) can be used to dynamically set and change configuration parameters of Internet hosts, including IP address, subnet mask, default router, and DNS server. DHCP is based on a client-server model. DHCP clients send requests to a DHCP server and the server responds with an allocation of IP addresses and other configuration parameters.

In this part of the lab, you will also learn about DHCP relay agents. When the DHCP client and DHCP server are not on the same IP network, DHCP relay agents can act as routers of DHCP messages. A DHCP relay agent can forward DHCP requests from a DHCP client to a DHCP server and it can forward the reply messages from the DHCP server to the DHCP client.

The network configuration for Part 2 is shown in Figure 2. PC1, PC3, and PC4 are set up as DHCP clients, and initially do not have IP addresses. PC2 is configured as a DHCP server, which listens for DHCP requests on all of its interfaces and transmits network configuration parameters. Router1 acts as a DHCP relay agent, which forwards DHCP messages between different IP networks.

Table 5 lists the range of addresses that are associated at the DHCP server PC2 with each IP network.

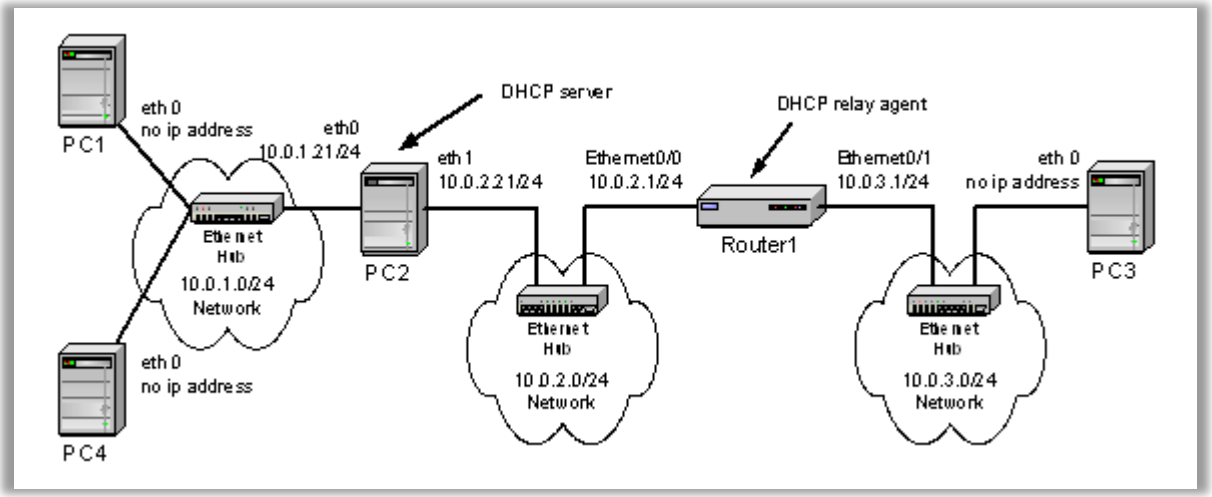


Figure 2. Network Configuration for Part 2.

Linux PC	IP Address of Internet Interface (eth0)	IP Address of Private Network Interface (eth1)	Default Gateway
PC1	None	none	none
PC2	10.0.1.21/24	10.0.2.21/24	10.0.2.1
PC3	None	none	none
PC4	None	none	None

Table 5. Configuration of the PCs in Part 2.

Cisco Router	IP Address of Interface (Ethernet0/0)	IP Address of Interface (Ethernet0/1)	Default Gateway
Router1	10.0.2.1/24	10.0.3.1/24	10.0.2.21

Table 6. Router configuration in Part 2.

Subnet	Range of Addresses	Default Router
10.0.1.0/24	10.0.1.2 to 10.0.1.10	10.0.1.21
10.0.3.0/24	10.0.3.2 to 10.0.3.10	10.0.3.1

Table 7. DHCP server configuration.

Exercise 2.a- Network Setup

Please do the following:

Step 1: Set up the network topology as shown in Figure 2. Configure the IP addresses of the PCs and Router1 as shown in Table 5 and Table 6.

Step 2: It is important that PC1, PC3 and PC4 do not have a default route and do not have an IP address associated with their respective interface eth0.

Review the routing table and the interface configuration. On PC1, this is done with the commands:

```
PC1$ netstat -nr
PC1$ ip address show
```

In Linux, if routing tables display the default route as an entry with destination '`0.0.0.0`', or '`default via ...`'. If the routing table shows a default route, you can delete this and all other routing table entries by issuing the following command:

```
PC1$ sudo ip route flush scope global
```

Exercise 2.b- Configuring and starting a DHCP server

On a Linux system, a DHCP server is started with the command `dhcpd`. The DHCP server reads the configuration file "`/etc/dhcp/dhcpd.conf`". The configuration file contains information on available IP addresses, and other configuration information. The following is an example of a configuration file for a DCHP server:

```
# dhcpd.conf file
ddns-update-style none;
default-lease-time 600;
subnet 10.0.1.0 netmask 255.255.255.0 {
```

```
   range 10.0.1.10 10.0.1.100;
   option routers 10.0.1.1;
}
subnet 10.0.2.0 netmask 255.255.255.0 {
   range 10.0.2.101 10.0.2.200;
}
subnet 10.0.3.0 netmask 255.255.255.0 {
   range 10.0.3.6 10.0.3.10;
   option routers 10.0.3.1;
}
```

The DHCP client is assigned an IP address for a period of time that is known as a *lease*. The above configuration file assigns IP addresses for a lease time of *600 seconds* (default-lease-time). For requests on network 10.0.1.0/24, the DHCP server assigns IP addresses in the range 10.0.1.10 – 10.0.1.100, assigns 10.0.1.1 as the default gateway. For requests on network 10.0.2.0/24, the server assigns IP addresses in the range 10.0.2.101– 10.0.2.200.

Step 1: **Copy the default configuration file:** Before making any changes to the `dhcp.conf` file, it is a good practice to copy it to another location so that you can retrieve it in case you make any inadvertent changes, use the following command:

```
PC2$ sudo cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.original
```

Step 2: **Set the DHCP configuration file:** On PC2, set up the configuration file so that IP addresses are assigned as follows. On PC2, using the sample configuration shown above, set up the configuration file so that IP addresses are assigned as follows:

- Set the default lease time to 120 seconds (you will see IP address lease renewal faster).
- On network 10.0.1.0/24, IP addresses are assigned in the range 10.0.1.2–10.0.1.10 with default gateway 10.0.1.21.
- On network 10.0.3.0/24, IP addresses are assigned in the range 10.0.3.2–10.0.3.10 with default gateway 10.0.3.1.
- Also, include the line

  ```
  subnet 10.0.2.0 netmask 255.255.255.0 {}
  ```

  This indicates that the DHCP server replies to requests that arrive on the interface belonging to subnet 10.0.2.0/24 (which is *eth1* on PC2). It also indicates that the DHCP server does not assign IP addresses on this subnet.
- The lease time of 600 seconds is the default value for assigning IP addresses. To make the renewal of IP address leases more frequent, replace the default lease time with the following lines:

  ```
  max-lease-time 120;
  min-lease-time 120;
  default-lease-time 120;
  ```

Step 3: On PC2, start a DHCP server with the command

```
PC2$ sudo service isc-dhcp-server start
```

Exercise 2.c- Starting a DHCP client

The following steps start a DHCP client on PC1.

Step 1: **Preparation:** On PC1, perform the following functions:

a. A Linux DHCP client caches information from previous uses of DHCP. The cached information is stored in files:

```
/var/lib/dhcp/dhclient.eth0.leases
/var/lib/dhcp/dhclient.eth0.leases~
```

Since this cached information may interfere with your work, delete these files, if they exist. Note that to erase these files you need to use 'sudo rm …'.

b. Start *Wireshark* to capture traffic on the interface eth0 of PC2. (Set the display filter to 'dhcp' so that only DHCP traffic is displayed in the window.)

Step 2: **Start a DHCP client:** The DHCP client on PC1 is started with the command

```
PC1$ sudo dhclient -d eth0
```

In the above command the option '-d' is used to run the process dhclient in the foreground and also to enable debugging to observe the process of acquiring an IP address. In general, the client process is run in the background, without interfering with the user's activities.

Explore the traffic captured by *Wireshark* to answer the following questions. (You may want to save the captured DHCP traffic to provide the answers for the lab report.)

a. Which IP address is assigned to PC1?

b. Describe the DHCP packets that are involved in the initial assignment of the IP address of PC1. For each packet include, the DHCP type, and the source and destination IPv4 addresses. Describe the role that each DHCP message type plays.

c. How is it possible that a host can send and receive DHCP packets, even though it does not have an IP address?

d. Do you observe any ARP packets? If so, explain the role of ARP in this context.

e. The DHCP client (PC1) sends a list of requested parameters to the DHCP server (PC2). The DHCP server provides some, but not all requested parameters. Provide a list of parameters requested by the client and the parameters that are provided by the serer.

Step 3: **Renewing leases of IP addresses:** The DHCP client is assigned an IP address for a limited period of time, which is called a lease. The maximum time of a lease is specified in the dhcpd.conf file. Information on current leases is stored at both the client side and the server side.

In Linux, information on the current leases is stored in the following files

At the DHCP server: `/var/lib/dhcp/dhcpd.leases`
At the DHCP client: `/var/lib/dhcp/dhclient.eth0.leases`
or: `/var/lib/dhcp/ dhclient.leases`

Inspect the content of the above files and the DHCP Request/DHCP ACK packets captured by Wireshark.

a. What type of DHCP messages can be observed during a lease renewal?

b. Compare the start and end times of the leases to the times when the DHCP client renews the lease. Provide three examples that show the start and end times of leases, and the times during the lease when the lease is renewed. Compare the lease length to the time interval after which the lease is renewed.

c. Stop the DHCP server with the command

```
PC2$ sudo service isc-dhcp-server stop
```

Observe the output on PC1 and the traffic capture of Wireshark to determine what the DHCP client does after the DHCP server is not reachable. Use the command `'ip addr show'` to determine how long the DHCP client waits until it releases the leased IP address.

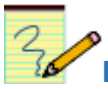
Step 4: Starting more DHCP clients:

Restart the DHCP server on PC2 with the command

```
PC2$ sudo service isc-dhcp-server restart
```

Use the instructions from Step 2 and start DHCP clients on PC3 and PC4.

a. The expected outcome is that PC4 receives an IP address, but that PC3 is not successful. Why is this outcome for PC3 expected?
b. Compare the IP addresses assigned to PC1 and PC4. Is there a specific order in which IP addresses are assigned by the DHCP server?



Lab Questions/Report

1. Provide answers to Step 2.
2. Provide answers to Step 3.
3. In most client-server application, the port number of a server is a well-known number (e.g., an *FTP* server uses port number 21, the telnet server uses port number 23, etc.), while the client uses a currently available (ephemeral) port number. DHCP is different. Here, both the client and

the server use a well-known port: UDP port 67 for the DHCP server, and UDP port 68 for the DHCP client. Refer to RFC 2131 and provide an explanation for this protocol design choice.

Exercise 2.d- DHCP relay agent

A DHCP relay agent can forward DHCP packets when the DHCP server and the DHCP client are not on the same network. Note that the role of a DHCP relay agent is not entirely trivial, since it acts as a router for a host that does not have an IP address. Here you explore, how packets from the client reach the server on another network, and how the response from the server reaches the DHCP client.

The DHCP server is configured to allocate addresses as shown in Table 7.

Step 1: **Setting up a Cisco router as a DHCP relay agent:** The following commands set up Router1 as a DHCP relay agent. In essence, Router1 is configured to forward UDP packets.

Start the DHCP relay agent on Router1 as follows. Note that the following lines are in addition to configuration statements which assign IP addresses to the interfaces:

```
Router1# configure terminal
Router1(config)# ip forward-protocol udp
Router1(config)# interface Ethernet0/1
Router1(config-if)# ip helper-address 10.0.2.21
Router1(config-if)# end
```

The following explains some of the commands.

IOS mode: Global configuration

`ip forward-protocol udp`
Enables UDP packet forwarding.

IOS mode: Interface configuration

`ip helper-address <IPaddr>`
Forwards DHCP request packets received on the current interface to IP address *<IPaddr>*, which should be the IP address of a DHCP server.

Step 2: Double-check that the default gateway at PC2 is set to IP address 10.0.2.1.

Step 3: Make sure that the DHCP server is running on PC2. If necessary, start a new DHCP server.

Step 4: Start *Wireshark* on PC2 (listening on *eth1*) and PC3 (listening on *eth0*).

Step 5: Start a DHCP client on PC3 with

```
PC3$ sudo dhclient -d eth0
```

Step 6: Verify that an IP address and a default router has been assigned to PC3.

Step 7: Answer the following questions.

a. Describe the changes that the DHCP relay server makes to IP destination and source addresses of DHCP packets.

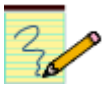
b. Describe the changes that the DHCP relay server makes to the fields of the BOOTP/DHCP message packets.

c. How does the relay agent redirect the replies from the DHCP server? Does it broadcast them or unicast them to the DHCP client?

d. How does the DHCP server (PC2) know on which network PC3 is located, when it receives the DHCP request?

Step 8: Save the DHCP packets that are captured by both *Wireshark* sessions (for use in the lab report). Then terminate all *Wireshark* traffic captures.

Lab Questions/Report



1. Provide the answers to the questions in Step 7.

2. Use the saved *Wireshark* data to provide a time-sequence diagram that shows the order of the DHCP packet transmissions between PC3, Router1, and PC2. Refer to Lab 3 (Part 5) for an example of a time-sequence diagram. For each DHCP packet, show the DHCP type, source and destination addresses, and the recorded time of transmission.

3. Combining NAT and DHCP

Figure 3 shows a network configuration which can be found in many SOHO (small office, home office) networks.

- The SOHO network is a private network with multiple hosts (PC1 and PC4) and one IP router (PC2).
- The IP router of the SOHO network (SOHO router) provides access to the public Internet by connecting to a router of an Internet service provider. The SOHO router obtains a single IP address on the "public" interface of the SOHO network via DHCP from a DHCP server (PC3) of the Internet service provider.
- The SOHO router works as a DHCP server and NAT server for the hosts in the SOHO network.

In this network setup, all SOHO hosts can share a single public IP address, which is dynamically assigned by the Internet service provider. Furthermore, the SOHO network requires minimal IP configuration. The hosts in the SOHO network obtain their IP address from the SOHO router. The SOHO router obtains its (public) IP address from the Internet service provider.

Your task is to setup the entire SOHO network, including the router and the DHCP server of the Internet service provider.

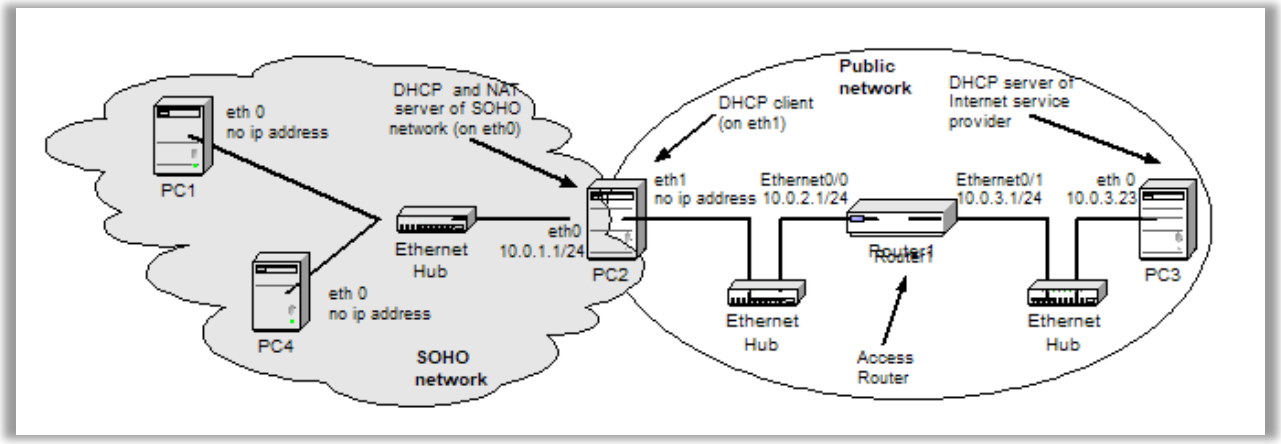


Figure 3. Network Configuration for Part 3.

Exercise 3:

The network configuration is shown as Figure 3. (The connections of the cables are identical to Figure 2). To reset the configuration of all machines, we recommend rebooting the PCs and the Cisco router.

Step 1: **DHCP Server:** PC3 is the DHCP server of the Internet service provider.

a. Configure PC3 with IP address 10.0.3.23/24 on interface eth0 and with default gateway 10.0.3.1.

b. Configurand start a DHCP server on PC3. On PC3, set up the configuration file so that IP addresses in the range 10.0.2.2-10.0.2.10 are assigned for requests on network 10.0.2.0/24,

with default gateway of 10.0.2.1, and addresses in the range 10.0.3.2-10.0.3.10 are assigned for requests on network 10.0.3.0/24, with default gateway of 10.0.3.1.

Step 2: **Router and DHCP relay agent:** Router1 is the IP router to which the SOHO network sends its external traffic. Also, Router1 is a DHCP relay agent.

- Configure Router1 with IP addresses 10.0.2.1/24 on interface Ethernet0/0 and 10.0.3.1/24 on interface Ethernet0/1.
- The routing table of Router1 should reflect that all traffic to network 10.0.2.0/24 is sent on interface Ethernet0/0, and all other traffic is sent on interface Ethernet0/1.
- Configure Router1 as a DHCP relay agent, so that requests from DHCP client PC2 reach DHCP server PC3.

Step 3: **SOHO Router:** Set up PC2 as the SOHO router with NAT.

- Set up PC2 so that it is a DHCP client on interface *eth1*.
- At this time, PC2 should acquire an IPv4 address for interface eth1. Verify this by viewing the IPv4 configuration or by issuing a ping from PC2 to PC3 with the commands

```
PC2$ ip addr show
PC2$ ping 10.0.3.23
```

  If this is not successful, debug the configuration. Refer to Table 8 below for hints on resolving potential issues.
- Set the IPv4 address of PC2 on eth0 to 10.0.1.1/24.
- Set up PC2 as an IPv4 router. That is, IPv4 forwarding must be enabled.
- Configure PC2 as a DHCP server on interface eth0 that assigns addresses in the range 10.0.1.2 – 10.0.1.10 with default gateway 10.0.1.1. Start a DHCP server process on PC2:

```
PC2$ sudo service isc-dhcp-server start
```

- Start a NAT server on PC2 and set up a NAT table, which maps packets from the SOHO network with source IP address from network 10.0.1.0/24 to the IP address of interface eth1, PC2 obtained through DHCP protocol from PC3. The command for adding a rule that will achieve this is:

```
PC2$ sudo iptables –t nat –A POSTROUTING –j MASQUERADE –o eth1 –s 10.0.1.0/24
```

Table 8. Hints when DHCP client on PC2 does not obtain an IPv4 address from DHCP server on PC3.

Problem	Hints
Use Wireshark on PC2 (eth1) and PC3 (eth0) to capture the traffic on both subnets of the public network.	
IP addresses on Router1 not configured or IP forwarding not enabled on Router1	If you issue a ping from PC3 to a non-existing IP address on subnet 10.0.2.0/24, for example, `PC3$ ping 10.0.2.20` Router1 should try to forward the ICMP Echo Request. Here, the Wireshark on PC2 (eth1) should capture ARP Requests for IP address 10.0.2.20. If you do not see this request, check the configuration of Router1.
DHCP relay agent not correctly configured on Router1	After PC2 starts its DHCP client, the Wireshark on PC3 (eth0) should capture DHCP Discover messages from PC2 that are relayed by Router1. If you do not see these messages, check the DHCP relay agent configuration on Router1. Note: Different from Part 2, in Part 3, the DHCP server is reachable via the Ethernet 0/1 interface of Router1.
DHCP server on PC3 not correctly configured	If the DHCP Discover messages reach PC3, but PC3 does not issue a DHCP Offer, the problem lies with the DHCP server on PC3. To make sure that the DHCP server is indeed running, restart the server with the command `PC3$ sudo service isc-dhcp-server restart` If the DHCP server is running, but no DHCP Offer is seen by Wireshark following a DHCP Discover message, check the `dhcpd.conf` following the instructions from Part 2.
DHCP client on PC2 not running	The DHCP client on PC2 gives up after sending a series of DHCP Discover messages without getting a DHCP Offer. If this is the case, start a new DHCP client.

Step 4: **Hosts in PCs:** PC1 and PC4 are hosts in the SOHO network.

- Set up PC1 and PC4 as DHCP clients on interfaces eth0.
- Now, PC1 and PC4 should acquire IPv4 addresses from the DHCP server on PC2. Also, PC1 and PC4 should be able to ping PC3. Verify this by viewing the IPv4 configuration of PC1 or by issuing a ping from PC1 to PC3 with the commands

```
PC1$ ip addr show
PC1$ ping 10.0.3.23
```

If this is not successful, debug the configuration. Refer to Table 9 below for hints on resolving potential issues.

Table 9. Hints when PC1 does not obtain an IPv4 address from PC2 or PC1 cannot communicate with PC3.

Problem	Hints
Start Wireshark to capture traffic on PC2 (eth0).	
IP addresses on PC2 (eth0) not configured	Confirm the IPv4 address of PC2 (eth0) with `PC2$ ip addr show`
DHCP server on PC2 not correctly configured	If PC2 does not issue a DHCP Offer after a DHCP Discover messages, the problem lies with the DHCP server on PC2. Restart the DHCP server with the command `PC2$ sudo service isc-dhcp-server restart` If there is still no DHCP Offer captured by Wireshark, check the `dhcpd.conf` file on PC2 following the instructions from Part 2.
DHCP client on PC1 or PC4 not running	The DHCP clients give up after sending a series of DHCP Discover messages without getting a DHCP Offer. If this is the case, start a new DHCP client.
IP forwarding not enabled on PC2	If PC1 has an IPv4 address, but PC1 cannot ping PC3, first check that IP forwarding is enabled on PC2. Then, confirm that PC2 can ping PC3, and consult Table 8 if this is not the case.
NAT not (correctly) configured on PC2	If IP forwarding is enabled on PC2, but NAT is not configured on PC2, then PC2 forwards ICMP Echo Requests from PC1 without modifying the source address of PC1. You can check this by running a Wireshark on PC2 (eth1). Also, check the NAT configuration on PC2 with the command `PC2$ sudo iptables –t nat -L` There should be an entry in the POSTROUTING chain. If there is no entry, check the `iptables` command from Step 3.

Step 5: Collect the results:

a. Display the routing tables from all PCs with '`netstat -nr`', and the IP configuration with '`ip addr show`', and take screen shots.

b. On Router1, display the routing table (`show ip route`) and the IPv4 configuration (show ip interfaces brief) and take a screenshot.

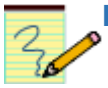
c. Display and save the NAT table of PC2. Take a screenshot of the output.

d. Take screenshots of the relevant lines (showing the assignment of addresses to subnets) of the `dhcpd.conf` files from PC2 and PC4.

e. Issue a `traceroute` from PC1 to PC3:

```
PC1$ traceroute  10.0.3.23
```

Take a screenshot of the output.



Lab Questions/Report

1. Provide the screenshots from Step 5. Present the screenshots in the order given in Step 10 and label each screenshot.