Based on
Mastering Networks - An Internet Lab Manual
by Jörg Liebeherr and Magda Al Zarki

Adapted for
'Labo Computernetwerken'
by Johan Bergs, Nicolas Letor, Michael Voorhaen and Kurt Smolderen

Completed by
Josse Coen Armin Halilovic Jonas Vanden Branden Group 2

May 16, 2016

Lab 7

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.

7.1 Prelab 7

NAT and DHCP

Use the following resources to prepare yourself for this lab session:

1. Unix commands for NAT, DHCP: Go to the online manual pages at http://manpages.ubuntu.com/. Read the manual pages of the following commands for the operating system version "trusty 14.04 LTS":
   - iptables
   - dhclient
   - dhcpd
   - dhcpd.conf
   - dhcp-options
   - dhcpd.leases
2. Private IP addresses: Read RFC 1918 on address allocation in private networks http://tools.ietf.org/html/rfc1918.
3. Network Address Translation (NAT): Read the following tutorial on NAT at http://www.firewall.cx/networking-topics/network-address-translation-nat.html.
4. Netfilter/iptables Read about netfilter and iptables at http://www.netfilter.org and http://www.thegeekstuff.com/2011/01/iptables-fundamentals/.
5. Dynamic Host Configuration Protocol (DHCP): Read RFC 2131 on DHCP at http://tools.ietf.org/html/rfc2131.

Prelab Questions

Question 1)

Explain why NAT is often mentioned as a solution to counteract the depletion of IP addresses on the global Internet? Which alternatives to NAT exist that address the scarcity of available IP addresses?

NAT is often mentioned as a solution to counteract the depletion of IP addresses on the global internet, because it allows different private networks to use the same IP addresses without creating conflicts between public networks.

There are specific blocks dedicated to those private networks (192.168/16, 10/8, 172.16/12). Everyone can use those in their own private network thanks to NAT.

An alternative to NAT is to just make longer IP addresses, which is exactly what IPv6 is doing.

Question 2)

What does the following comment refer to: “NAT destroys the ability to do host-to-host communication over the Internet“?

IP was originally supposed to be an end-to-end protocol, where each IP address uniquely identifies a host. Any host would then be able to address any other host using just its IP address, which is the host-to-host communication the comment refers to. With NAT and IP masquerading however, a host may be “hidden” behind a network address translator (i.e. a router with NAT configured), along with other hosts behind the same router. As such a host can’t uniquely identify a host using an IP address anymore. More information is needed, such as a port number. The NAT protocol changes the IP addresses when resolving to the host in the local network. This way, the exact IP address is not used in the full communication process and the NAT-router intervenes with IP datagrams.

Explain the following terms which are used in the context of Network Address Translation:

Question 3.a)

Static NAT

The NAT maps a certain private host with a static public IP address, meaning that this host will always use this IP address when communicating with the public network.

Question 3.b)

Dynamic NAT

With dynamic NAT, the NAT router has a pool of public IP‘s from which it can choose one to assign to a private host when he wants to access the public internet.

Question 3.c)

NAT with IP overload

NAT overload makes the hosts share public IPs, as this overcomes the limitation of static and dynamic NAT, where the number of hosts is limited by the amount of available public IP addresses.

To separate the different datastreams from the different host, a different src port is used for each stream.

Question 3.d)

Port Address Translations e.g. IP Masquerading

With IP masquerading, different hosts share a public IP address to access the internet.

Masquerading is essentially a many-to-one mapping where only one public IP address is mapped to many private IP addresses.

Question 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?
Section 3 (Private Address Space) lists the following blocks:

10.0.0.0 - 10.255.255.255 (10/8 prefix)
172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

If there were no blocks specified for private networks specifically (so private networks are allowed to use a network prefix that also exists publicly), there could be ambiguity as to whether a packet sent from within a private network is destined for a host in the private network or the public network.

Question 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?
Netfilter is the internal packet-manipulation framework used by the kernel. IPtables is the user interface which manipulates this framework.

Describe the following terms which are used in the iptables command:

Question 6.a)

Chain
From the iptables manual page:
Each chain is a list of rules which can match a set of packets.
Each rule specifies what to do with a packet that matches.
This is called a 'target', which may be a jump to a user-defined chain in the same table.

Question 6.b)

Postrouting
Packets are altered when they are about to go out.

Question 6.c)

Prerouting
Packets are altered as soon as they come in.

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.

Question 7.a)

Write the iptables command so that the addresses in the private network are mapped to the public IP address 128.143.136.80.

```
iptables -t nat -A POSTROUTING -j SNAT –to 128.143.136.80 -s 10.0.1.0/24
```

Question 7.b)

Write an IOS command so that the addresses in the private network are mapped to the public IP address 128.143.136.80.

```
ip nat inside source static IPAddress 128.143.136.80
```

Answer the following questions about DHCP:

Question 8)

Explain the meaning of the “magic cookie” in the DHCP protocol.

The magic cookie is a certain value of the first four bytes of the options field in a DHCP packet, indicating that the information given is vendor-independent.

Question 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?

dhcpd, when issued without arguments, causes the DHCP server to listen on all network interfaces.

7.2 Lab 7

Figure 7.1 shows two private networks 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.

In the private networks in Figure 7.1, Router1 and Router3 are used to mimic hosts, i.e., they are not configured to act as IP routers.)

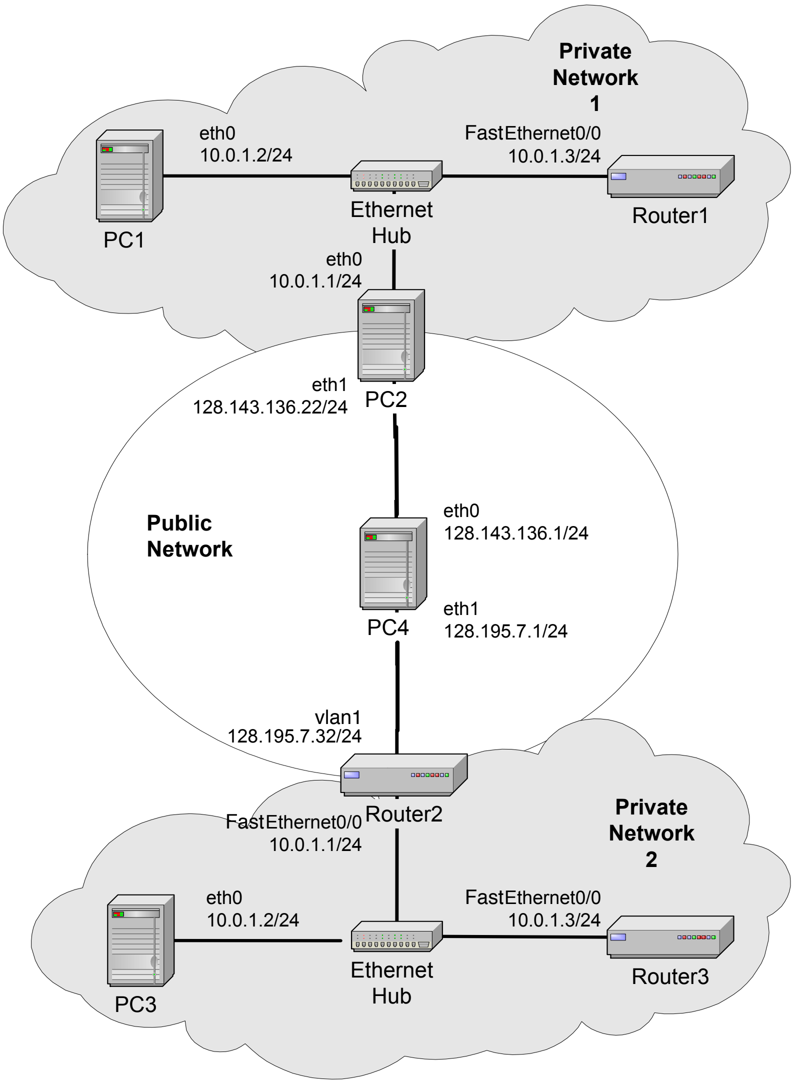


Figure 7.1: Network configuration for Part 1.

- In this lab, PC2 and Router2 are routers that provide the gateways between the private and the public networks. Both PC2 and Router2 are configured as NAT devices.
- On PC2, the kernel is built with netfilter, an extension to the Linux kernel that provides the kernel with the ability to set IP packet filters, including NAT functions. On Router2, you will

Linux PC	IP Addresses of eth0	IP Addresses of 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
PC3	128.143.136.1/24	128.195.7.1/24	none

Table 7.1: IP addresses and gateways assignment of all PCs for Part 1.

use Cisco IOS commands to configure NAT rules.

- PC4 runs as an IP router. (We use a Linux PC instead of a Cisco Router so that wireshark can be used to capture traffic on the public network).
- The assignment of IP addresses and default gateways for all PCs and routers are shown in Table1 and Table 2.
- The console port of Router1 is connected to a serial port of PC1, the console port of Router2 is connected to a serial port of PC2, and the console port of Router3 is connected to a serial port of PC3.

Linux PC	IP Addresses of FA0/0	IP Addresses of vlan1	Default Gateway	Connected PC
Router1	10.0.1.3/24	none	10.0.1.1	PC1
Router2	10.0.1.1/24	128.195.7.32/24	128.195.7.1	PC2
Router3	10.0.1.3/24	none	10.0.1.1	PC3

Table 7.2: IP addresses and gateways assignment of all routers for Part1.

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 7.1, 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 7.1 with the IP address configuration shown in Table 7.1 and Table 7.2. The following commands review the steps involved in the configuration.

1. On the Linux PCs, use `ifconfig` to configure the IP address of the interfaces. Add a default gateway on each PC with the command (shown for PC1):

   ```
   PC1% route add default gw gateway_address
   ```

2. IP forwarding must be enabled on PC2 and PC4.

3. Use a serial cable to connect a serial port of a PC to the console port of a router. Use the `minicom` command to access the routers.

4. Configure the IP addresses of interfaces *Fa/0* and *vlan1* on the routers, and set the default gateways as shown in Table 7.2. Below is the sample configuration for Router2.

   ```
   Router2> enable
   Password: <enable secret>
   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 FastEthernet0/0
   Router2(config-if)# no shutdown
   Router2(config-if)# ip address 10.0.1.1 255.255.255.0
   Router2(config-if)# interface FastEthernet0/1
   Router2(config-if)# no shutdown
   Router2(config-if)# interface vlan1
   Router2(config-if)# no shutdown
   Router2(config-if)# ip address 128.195.7.32 255.255.255.0
   Router2(config-if)# end
   ```

The following commands sets 128.195.7.1 as the default gateway of Router2.

```
Router2(config)# ip route 0.0.0.0 0.0.0.0 128.195.7.1
```

After completing the set up of the configuration you should be able to issue successful intra network ping commands i.e., 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.

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

You will use Wireshark in this exercise. Do not forget to append the binary dump (pcap format) to your lab report

A Cisco router can be set up to run as a NAT device.

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 that is not made known in the private network.*
- *An outside local address is used by devices in the private network to addresses 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.

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

   - A NAT rule is added so that the private IP address of PC3, 10.0.1.2, is translated to the public address 200.0.0.2. The IOS commands are as follows:

```
Router2> enable
Password: <enable secret>
Router2# show ip nat translations
Router2# configure terminal
Router2(config)# interface FastEthernet0/0
Router2(config-if)# ip nat inside
Router2(config-if)# interface vlan1
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
Router2# show ip nat translations
```

- After the above rule has been entered, display the content of the NAT table and save it to a file. The commands used above are explained below:
  - Displays the content of the NAT table:

    ```
    Router2# show ip nat translations
    ```
  - Specifies that interface *FastEthernet0/0* is connected to the private network.

    ```
    Router2(config)# interface FastEthernet0/0
    Router2(config-if)# ip nat inside
    ```
  - Specifies that interface *vlan1* is connected to the public network.

    ```
    Router2(config-if) #interface vlan1
    Router2(config-if)# ip nat outside
    ```
  - Adds a rule so that the private address 10.0.1.2 is mapped to the public address 200.0.0.2

    ```
    Router2(config)# ip nat inside source static 10.0.1.2 200.0.0.2
    ```

"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.

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

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 an Wireshark session on PC3 to capture the traffic from Router2 on the private network.
   - Start an 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 -c 3 10.0.1.3
     PC3% ping -c 3 128.143.136.1
     ```

     On Router3:

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

     On PC4:

     ```
     PC4% ping -c 3 10.0.1.2
     PC4% ping -c 3 200.0.0.2
     ```
   - Save the Wireshark data to files. Observe which ping commands succeed.

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 7.3.

Linux PC	Inside local address	Outside local address
Router2	10.0.1.1/24	200.0.0.1
Router3	10.0.1.3/24	200.0.0.3

Table 7.3: Private and public addresses of Router2 and Router3.

Question 1.B.a)

Include the NAT table of Router2 and provide an explanation of the columns of the table.

```
Pro  Inside global      Inside local      Outside local      Outside global
---  200.0.0.1          10.0.1.1          ---                ---
---  200.0.0.2          10.0.1.2          ---                ---
---  200.0.0.3          10.0.1.3          ---                ---
```

Pro

Protocol of the port identifying the address

Inside local

An IP address assigned to a host on the private network.

Inside global

An IP address that represents one or more inside local IP addresses to the outside world.

Outside local

The IP address of an outside host as it appears to the inside network.

Outside global

The IP address assigned to a host on the outside network by the owner of the host. The address is allocated from a globally routable address or network space.

Question 1.B.b)

For each of the ping commands above, provide an explanation why the command succeeds or fails.

```
(1) PC3% ping -c 3 10.0.1.3  : SUCCESS
(2) PC3% ping -c 3 128.143.136.1: SUCCESS

(3) Router3# ping 10.0.1.2   : SUCCESS
(4) Router3# ping 128.143.136.1  : SUCCESS

(5) PC4% ping -c 3 10.0.1.2  : UNREACHABLE
(6) PC4% ping -c 3 200.0.0.2   : SUCCESS
```

1. Pings to local Router3.
2. Router2 is PC3's default gateway, Request is sent to Router2.
   Request's source IP is changed to 200.0.0.2.
   PC4 is Router2's default gateway, Request is sent to PC4.
   On PC4, traffic with destination IP address 200.0.0.0/24 is forwarded to Router2, so Reply is sent to Router2.
   Reply's destination IP is changed to 10.0.1.2.
   Reply is sent to PC3.
3. Pings to local PC3.
4. Analogous to (2).
5. PC4 cannot reach private address 10.0.1.2. 10.0.1.0/24 does not match anything in its routing table.
6. On PC4, traffic with destination IP address 200.0.0.0/24 is forwarded to Router2, so Request is sent to Router2.
   Request's destination IP is changed to 10.0.1.2.
   Request is sent to PC3.

Router2 is PC3's default gateway, Reply is sent to Router2.
Reply's source IP is changed to 200.0.0.2.
PC4 is Router2's default gateway, Reply is sent to PC4.

Question 1.B.c)

Include 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.

ICMP request before Router2 (/Lab 7/traces/1.B-3.PC3.pcap):

```
No.     Time          Source                Destination           Protocol Length
    Info
     26 61.288832     10.0.1.2              128.143.136.1         ICMP     98
        Echo (ping) request  id=0x032c, seq=2/512, ttl=64 (reply in 27)

Frame 26: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr 11, 2016 11:37:30.594998000 CEST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1460367450.594998000 seconds
    [Time delta from previous captured frame: 0.998787000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 61.288832000 seconds]
    Frame Number: 26
    Frame Length: 98 bytes (784 bits)
    Capture Length: 98 bytes (784 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:icmp:data]
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
Ethernet II, Src: IntelCor_36:39:c7 (68:05:ca:36:39:c7), Dst: Cisco_ef:b1:a3 (00:0d
    :bc:ef:b1:a3)
    Destination: Cisco_ef:b1:a3 (00:0d:bc:ef:b1:a3)
        Address: Cisco_ef:b1:a3 (00:0d:bc:ef:b1:a3)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: IntelCor_36:39:c7 (68:05:ca:36:39:c7)
        Address: IntelCor_36:39:c7 (68:05:ca:36:39:c7)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IP (0x0800)
Internet Protocol Version 4, Src: 10.0.1.2 (10.0.1.2), Dst: 128.143.136.1
    (128.143.136.1)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (
        Not ECN-Capable Transport))
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable
            Transport) (0x00)
    Total Length: 84
    Identification: 0x6d44 (27972)
    Flags: 0x02 (Don't Fragment)
        0... .... = Reserved bit: Not set
        .1.. .... = Don't fragment: Set
        ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (1)
    Header checksum: 0xb9d2 [validation disabled]
        [Good: False]
        [Bad: False]
    Source: 10.0.1.2 (10.0.1.2)
    Destination: 128.143.136.1 (128.143.136.1)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xa923 [correct]
    Identifier (BE): 812 (0x032c)
    Identifier (LE): 11267 (0x2c03)
    Sequence number (BE): 2 (0x0002)
    Sequence number (LE): 512 (0x0200)
    [Response frame: 27]
    Timestamp from icmp data: Apr 11, 2016 11:37:30.000000000 CEST
    [Timestamp from icmp data (relative): 0.594998000 seconds]
    Data (48 bytes)

0000  1e 14 09 00 00 00 00 00 10 11 12 13 14 15 16 17   ................
0010  18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27   ........ !"#$%&'
0020  28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37   ()*+,-./01234567
        Data: 1e14090000000000101112131415161718191a1b1c1d1e1f...
        [Length: 48]
```

ICMP request after Router2 (/Lab 7/traces/1.B-3.PC4.pcap):

```
No.     Time          Source                Destination           Protocol Length
    Info
     22 37.045516     200.0.0.2             128.143.136.1         ICMP     98
        Echo (ping) request  id=0x032c, seq=2/512, ttl=63 (reply in 23)

Frame 22: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr 11, 2016 11:37:29.007583000 CEST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1460367449.007583000 seconds
    [Time delta from previous captured frame: 0.997630000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 37.045516000 seconds]
    Frame Number: 22
    Frame Length: 98 bytes (784 bits)
    Capture Length: 98 bytes (784 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:icmp:data]
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
Ethernet II, Src: Cisco_ef:b1:a3 (00:0d:bc:ef:b1:a3), Dst: IntelCor_39:e1:32
    (68:05:ca:39:e1:32)
    Destination: IntelCor_39:e1:32 (68:05:ca:39:e1:32)
        Address: IntelCor_39:e1:32 (68:05:ca:39:e1:32)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: Cisco_ef:b1:a3 (00:0d:bc:ef:b1:a3)
        Address: Cisco_ef:b1:a3 (00:0d:bc:ef:b1:a3)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IP (0x0800)
Internet Protocol Version 4, Src: 200.0.0.2 (200.0.0.2), Dst: 128.143.136.1
    (128.143.136.1)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (
        Not ECN-Capable Transport))
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable
            Transport) (0x00)
    Total Length: 84
    Identification: 0x6d44 (27972)
    Flags: 0x02 (Don't Fragment)
        0... .... = Reserved bit: Not set
        .1.. .... = Don't fragment: Set
        ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 63
    Protocol: ICMP (1)
    Header checksum: 0xfdd1 [validation disabled]
        [Good: False]
        [Bad: False]
    Source: 200.0.0.2 (200.0.0.2)
    Destination: 128.143.136.1 (128.143.136.1)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xa923 [correct]
    Identifier (BE): 812 (0x032c)
    Identifier (LE): 11267 (0x2c03)
    Sequence number (BE): 2 (0x0002)
    Sequence number (LE): 512 (0x0200)
    [Response frame: 23]
    Timestamp from icmp data: Apr 11, 2016 11:37:30.000000000 CEST
    [Timestamp from icmp data (relative): -0.992417000 seconds]
    Data (48 bytes)

0000  1e 14 09 00 00 00 00 00 10 11 12 13 14 15 16 17   ................
0010  18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27   ........ !"#$%&'
0020  28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37   ()*+,-./01234567
        Data: 1e14090000000000101112131415161718191a1b1c1d1e1f...
        [Length: 48]
```

Corresponding reply before Router2 (/Lab 7/traces/1.B-3.PC4.pcap):

```
No.     Time          Source                Destination           Protocol Length
    Info
     23 37.045555     128.143.136.1         200.0.0.2             ICMP     98
        Echo (ping) reply    id=0x032c, seq=2/512, ttl=64 (request in 22)

Frame 23: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr 11, 2016 11:37:29.007622000 CEST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1460367449.007622000 seconds
    [Time delta from previous captured frame: 0.000039000 seconds]
    [Time delta from previous displayed frame: 0.000039000 seconds]
    [Time since reference or first frame: 37.045555000 seconds]
    Frame Number: 23
    Frame Length: 98 bytes (784 bits)
    Capture Length: 98 bytes (784 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:icmp:data]
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
Ethernet II, Src: IntelCor_39:e1:32 (68:05:ca:39:e1:32), Dst: Cisco_ef:b1:a3 (00:0d
    :bc:ef:b1:a3)
    Destination: Cisco_ef:b1:a3 (00:0d:bc:ef:b1:a3)
        Address: Cisco_ef:b1:a3 (00:0d:bc:ef:b1:a3)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: IntelCor_39:e1:32 (68:05:ca:39:e1:32)
        Address: IntelCor_39:e1:32 (68:05:ca:39:e1:32)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IP (0x0800)
Internet Protocol Version 4, Src: 128.143.136.1 (128.143.136.1), Dst: 200.0.0.2
    (200.0.0.2)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (
        Not ECN-Capable Transport))
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable
            Transport) (0x00)
    Total Length: 84
    Identification: 0x579c (22428)
    Flags: 0x00
        0... .... = Reserved bit: Not set
        .0.. .... = Don't fragment: Not set
        ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (1)
    Header checksum: 0x527a [validation disabled]
        [Good: False]
        [Bad: False]
    Source: 128.143.136.1 (128.143.136.1)
    Destination: 200.0.0.2 (200.0.0.2)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0xb123 [correct]
    Identifier (BE): 812 (0x032c)
    Identifier (LE): 11267 (0x2c03)
    Sequence number (BE): 2 (0x0002)
    Sequence number (LE): 512 (0x0200)
    [Request frame: 22]
    [Response time: 0,039 ms]
    Timestamp from icmp data: Apr 11, 2016 11:37:30.000000000 CEST
    [Timestamp from icmp data (relative): -0.992378000 seconds]
    Data (48 bytes)

0000  1e 14 09 00 00 00 00 00 10 11 12 13 14 15 16 17   ................
0010  18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27   ........ !"#$%&'
0020  28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37   ()*+,-./01234567
        Data: 1e14090000000000101112131415161718191a1b1c1d1e1f...
        [Length: 48]
```

Corresponding reply after Router2 (/Lab 7/traces/1.B-3.PC3.pcap):

```
No.     Time         Source                Destination           Protocol Length
    Info
     27 61.290324    128.143.136.1         10.0.1.2              ICMP     98
        Echo (ping) reply    id=0x032c, seq=2/512, ttl=63 (request in 26)

Frame 27: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr 11, 2016 11:37:30.596490000 CEST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1460367450.596490000 seconds
    [Time delta from previous captured frame: 0.001492000 seconds]
    [Time delta from previous displayed frame: 0.001492000 seconds]
    [Time since reference or first frame: 61.290324000 seconds]
    Frame Number: 27
    Frame Length: 98 bytes (784 bits)
    Capture Length: 98 bytes (784 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:icmp:data]
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
Ethernet II, Src: Cisco_ef:b1:a3 (00:0d:bc:ef:b1:a3), Dst: IntelCor_36:39:c7
    (68:05:ca:36:39:c7)
    Destination: IntelCor_36:39:c7 (68:05:ca:36:39:c7)
        Address: IntelCor_36:39:c7 (68:05:ca:36:39:c7)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: Cisco_ef:b1:a3 (00:0d:bc:ef:b1:a3)
        Address: Cisco_ef:b1:a3 (00:0d:bc:ef:b1:a3)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IP (0x0800)
Internet Protocol Version 4, Src: 128.143.136.1 (128.143.136.1), Dst: 10.0.1.2
    (10.0.1.2)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (
        Not ECN-Capable Transport))
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable
            Transport) (0x00)
    Total Length: 84
    Identification: 0x579c (22428)
    Flags: 0x00
        0... .... = Reserved bit: Not set
        .0.. .... = Don't fragment: Not set
        ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 63
    Protocol: ICMP (1)
    Header checksum: 0x107b [validation disabled]
        [Good: False]
        [Bad: False]
    Source: 128.143.136.1 (128.143.136.1)
    Destination: 10.0.1.2 (10.0.1.2)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0xb123 [correct]
    Identifier (BE): 812 (0x032c)
    Identifier (LE): 11267 (0x2c03)
    Sequence number (BE): 2 (0x0002)
    Sequence number (LE): 512 (0x0200)
    [Request frame: 26]
    [Response time: 1,492 ms]
    Timestamp from icmp data: Apr 11, 2016 11:37:30.000000000 CEST
    [Timestamp from icmp data (relative): 0.596490000 seconds]
    Data (48 bytes)

0000  1e 14 09 00 00 00 00 00 10 11 12 13 14 15 16 17   ................
0010  18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27   ........ !"#$%&'
0020  28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37   ()*+,-./01234567
        Data: 1e1409000000000010111213141516171819 1a1b1c1d1e1f...
        [Length: 48]
```

Exercise 1-c: IP Masquerading with a Linux PC

You will use Wireshark in this exercise. Do not forget to append the binary dump (pcap format) to your lab report

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.

In this exercise, PC2 will be configured to perform IP masquerading. The Linux kernel on all PCs has been built with netfilter, 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.

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% iptables -t nat
```

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

`iptables -t nat -L`

> *Displays all rules in the NAT table*

`iptables -t nat -L`

> *Deletes the first rule in the POSTROUTING chain of the NAT table*

`iptables -t nat -F`

> *Deletes all entries in ("flushes") the NAT table*

`iptables -t nat -A POSTROUTING -j SNAT --to IPAddr -s PrivateIPAddr/netmask`

> *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 PrivateIPAddr/netmask is changed to IPAddr".*
> *Example: The source address of outgoing IP datagrams that match "10.0.1.0/24" is changed to 128.195.7.32.*
> *`iptables -t nat -A POSTROUTING -j SNAT --to 128.195.7.32 -s 10.0.1.0/24`*

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 save it to a file.

2. Observe traffic at a NAT device:

   - 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*.
   - Establish a set of Telnet session and login to remote machines, using the following `telnet` commands: On PC1:

     ```
     PC1% telnet 10.0.1.3
     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
     ```

   - Save the Wireshark data to files. Observe which Telnet commands succeed.
   - For the successful Telnet sessions, observe how the IP addresses and port numbers are mapped.

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.

   - On PC2, run Wireshark on both *eth0* and *eth1*. Use the appropriate filters to capture the traffic generated by ping commands.
   - Issue the following ping commands: On PC1:

```
PC1% ping -c 3 10.0.1.3
PC1% ping -c 3 128.143.136.1
```

On Router1:

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

On PC4:

```
PC4% ping -c 3 10.0.1.2
```

- Save the Wireshark output and the output of ping commands into files.

Question 1.C.a)

For each of the `telnet` and `ping` commands above, provide an explanation why a command succeeds or fails.

TELNET

```
(1) PC1% telnet 10.0.1.3      : SUCCESS
(2) PC1% telnet 128.143.136.1    : SUCCESS

(3) Router1# telnet 10.0.1.2      : SUCCESS
(4) Router1# telnet 128.143.136.1  : SUCCESS

(5) PC4% telnet 10.0.1.2      : UNREACHABLE
```

PINGS

```
(1) PC1% ping -c 3 10.0.1.3     : SUCCESS
(2) PC1% ping -c 3 128.143.136.1   : SUCCESS

(3) Router1# ping 10.0.1.2      : SUCCESS
(4) Router1# ping 128.143.136.1    : SUCCESS

(5) PC4% ping -c 3 10.0.1.2     : UNREACHABLE
```

For both, the following explanation holds:

1. In same private network, to local IP (Router1)
2. To public network, using the recently added NAT chain rule on PC2.
3. In same private network, to local IP (PC1)
4. To public network, using the recently added NAT chain rule on PC2.
5. Can't find this ip in the public network.

Question 1.C.b)

For each successful telnet session, include the IP header data of an outgoing and an incoming packet header (with respect to the private network).

*** Missing wireshark files ***

Question 1.C.c)

For each successful ping command, include the IP header data of an outgoing ICMP Request message and an incoming ICMP reply message (with respect to the private network).

*** Missing wireshark files ***

Question 1.C.d)

How does PC know that a packet coming from the public network is destined to a host in the private network?

In case of Telnet, by the destination's port number in the TCP header. For the pings, by the value in the Identifier field. When an inside host reaches out to the outside, the translator stores these so that when it receives a packet from the outside directed to itself, it knows which private destination IP address to translate to.

Question 1.C.e)

Explain the steps performed by the kernel during IP address translation.

For a packet from the inside destined to the outside, the port number (or other kinds of identification, as mentioned, depends on the protocol) is mapped to the private address of the host, in order to track the connection. The packet's source IP address is then replaced by the router's outside IP address. Because of this, responses from the outside will be directed towards the router.

When the router receives a packet from the outside, it looks up the port number or ICMP identifier or ... (again, depends on the protocol) and, if found, translates the packet's destination IP address field to the corresponding IP address.

Exercise 1-d: NAT and FTP

You will use Wireshark in this exercise. Do not forget to append the binary dump (pcap format) to your lab report

NAT can create problems for applications, which carry the IP addresses in the payload of an IP datagram. An example of such an application is the file transfer program (FTP).

In this exercise, you establish an FTP connection from PC3 in the private network to PC2 in the public network, and observe how the FTP application works with NAT.

1. Start Wireshark on interface *eth0* of PC4 and on interface *eth0* of PC3.
2. FTP session between two hosts in the public network:
   - Start the FTP server on PC2 by typing
     ```
     PC2% service vsftpd start
     ```
   - Start an FTP connection from PC4 to PC2 (the -d option prints out debug messages).
     ```
     PC4% cd /root/labdata
     PC4% ftp -d 128.143.136.22
     ```
     Login with user name "root" and enter the root password.
   - Download a file from the FTP server.
     ```
     ftp> get fname
     ```
     where `fname` is a file on the remote server. (You can use the command ls to obtain a list of all files in the remote directory.)
   - Use the traffic captured by Wireshark to determine where the payload of FTP data carries information on IP addresses.
   - Save the Wireshark output and the FTP debug information output into files.
3. FTP session from a private to the public network:
   - Use the same commands as previously to download a file from PC2 to PC3

```
PC3% ftp -d 128.143.136.22
```

Is the FTP session establishment successful?

- Save the traffic captured by wireshark and save the FTP debug information output. Make sure that you save enough data to answer the lab report questions.

Question 1.D.a)

Use the captured data to explain the outcome of the FTP experiment. In particular, if the file was successfully downloaded, explain how the problem of sending the IP address as part of the data payload of the IP packet is solved.

When downloading a file from an FTP server, we see that a PORT command is sent by the client before the transfer starts. This is where the payload of FTP data carries information on IP addresses.

A client sends a PORT command to an FTP server to set up active mode. In active mode, an FTP server will initiate a connection to the client, instead of waiting for a connection attempt from the client. It can connect to the client, because the client has specified the address and port number it is listening on in the PORT command.

Examples of PORT commands: packet no. 48 in “/Lab 7/traces/1.D.PC3.pcap”, packet no. 29 and 79 in “/Lab 7/traces/1.D.PC4.pcap”

We note that packet 48 is sent by PC3 and is captured before going through Router2. Packet 79 is this same packet after going through Router2.

We can see that after the address translation, the address has also been changed in the FTP payload. This is possible because the network address translator knows about the FTP protocol, so it also knows where to find IP addresses in the payload. In addition of translating the IP addresses in the IP header, it will also translate those found in the payload. This, of course, won’t work with any protocols that the translator doesn’t know.

In our case, the FTP-DATA has destination 200.0.0.2:51441, this goes to Router2, and then is sent to PC3.

Question 1.D.b)

How can NAT be used to spoof a host address? How can you prevent this?

NAT could be used to change the source address of outgoing IP datagrams to the address of another host.

*** How can you prevent this? ***

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 7.2. 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 7.4 lists the range of addresses that are associated at the DHCP server PC2 with each IP network.

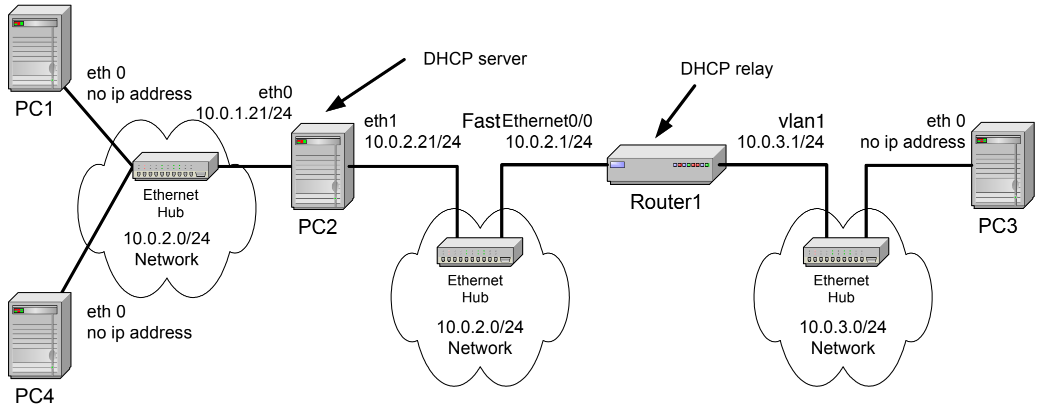


Figure 7.2: Network configuration for Part 2.

Linux PC	IP Addresses of eth0	IP Addresses of 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
PC3	none	none	none

Table 7.4: Configuration of the PCs in Part 2.

Linux PC	IP Addresses of eth0	IP Addresses of eth1	Default Gateway	Connected PC
Router1	10.0.2.1/24	10.0.3.1/24	10.0.2.21	PC1

Table 7.5: Configuration of the Routers 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 7.6: DHCP server configuration.

Exercise 2-a: Network Setup

1. We strongly recommend that you reboot the PCs and the routers before you proceed. Don't forget to save your files on a USB stick or online before rebooting.
2. Set up the network topology as shown in Figure 7.2. Configure the IP addresses of the PCs and Router1 as shown in Table 7.4 and Table tab:lab7-part2-ip-addresses-routers.
3. 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 -rn
PC1% ifconfig -a
```

In Linux, routing tables display the default route as an entry with destination 0.0.0.0. If the routing table shows a default route, you can delete this and all other routing table entries by setting the IP address to 0.0.0.0. This is done with the following command:

```
PC1% ifconfig eth0 0.0.0.0 up
```

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/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
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;
        default-lease-time 120;
}
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;
}
```

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, and limits the lease of addresses to 120 seconds, thus, overruling the global limit of 600 seconds. 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.

1. Set the DHCP configuration file: On PC2, set up the configuration file so that IP addresses are assigned as follows. 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. Note that these assignments are similar to, but not identical with the configuration file shown above.
2. Start the DHCP server: On PC2, start the DHCP server by typing

   ```
   PC2% dhcpd
   ```

   The DHCP server daemon listens for requests from DHCP clients on all its interfaces. In Linux, the DHCP server must be restarted each time the configuration file is modified. Since only one DHCP server can run at a time, you may need to terminate the current DHCP server process.

Exercise 2-c: Starting a DHCP client

You will use Wireshark in this exercise. Do not forget to append the binary dump (pcap format) to your lab report

The following steps start a DHCP client on PC1.

1. On PC1, perform the following functions:
   - Ensure that no default router entry exists in the routing table.
   - A Linux DHCP client caches information from previous uses of DHCP. The cached information is stored in :

     ```
     /var/lib/dhcp3/
     ```

     Since this cached information may interfere with your work, delete the lease files related to dhclient, if they exist:

     ```
     rm /var/lib/dhcp3/dhclient*
     ```
   - Start Wireshark on interface *eth0* of PC2. (Set the display filter to "bootp.dhcp" so that only DHCP traffic is displayed in the window.)
2. Start a DHCP client with the command

   ```
   PC1% dhclient eth0
   ```

Save the data that is captured by Wireshark to a file. Save enough data to answer the following questions from the captured traffic:

Question 2.C.2.a)

Which IP address is assigned to PC1?

10.0.1.2

Question 2.C.2.b)

Observe the source and destination IP addresses of the packets that are sent between DHCP client and DHCP server.

from client:
src: 0.0.0.0 dst: 255.255.255.255
src: 10.0.1.2 dst: 10.0.1.21

from server:
src: 10.0.1.21 dst: 10.0.1.2

Question 2.C.2.c)

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

The client can broadcast DHCP packets with:
source hardware address: its own hardware address
source ip address: 0.0.0.0

destination hardware address: ff:ff:ff:ff:ff:ff
destination ip address: 255.255.255.255

The client can then be addressed/identified by the server, using the client's hardware address.
The DHCP server will know this address because of the client's broadcast message.

Question 2.C.2.d)

Do you observe any ARP packets? If so, explain the function of the ARP in this context.
Yes.
After receiving an IP address the DHCP server still would need to configure its ARP table.
This could be interpreted as a confirmation of the DHCP assignment process.

Question 2.C.2.e)

Observe and interpret the output of the DHCP packets. You should see the following packet types: DHCP Discover, DHCP Offer, DHCP Request, DHCP ACK.

DHCP Discover:
Broadcasted by a DHCP client to request a DCHP offer packet from a DHCP server on the network.

DHCP Offer:
Unicast from DHCP server to DHCP client. Contains the available IP address and other information (such as subnet mask, lease time, default gateway, etc.) that the DHCP server offers to the DHCP client.

DHCP Request:
Broadcast from client to server. The client requests the server to assign the offered IP address to it.

DHCP ACK:
Unicast from server to client. The server acknowledges the assignment of the offered IP address for a specific lease time.

We note that the Offer and ACK packets contain exactly the same information, except for the Message Type.

Question 2.C.2.f)

Identify and interpret all option fields in the DHCP packet types that you observe.

(1) Subnet Mask
example data:

```
    Length: 4
    Subnet Mask: 255.255.255.0 (255.255.255.0)
```

The subnet mask set by a server for a client.

(12) Host Name

example data:

```
    Length: 7
    Host Name: lab2pc1
```

The host name of the sender.

(3) Router

example data:

```
    Length: 4
    Router: 10.0.1.21 (10.0.1.21)
```

List of available routers.

(50) Requested IP Address

example data:

```
    Length: 4
    Requested IP Address: 10.0.1.2 (10.0.1.2)
```

Set by the client to request a specific IP address.

(51) IP Address Lease Time

example data:

```
    Length: 4
    IP Address Lease Time: (600s) 10 minutes
```

The lease time for the IP assignment, set by the server.

(53) DHCP Message Type

example data:

```
    Length: 1
   DHCP: Discover (1)
```

The type of the DHCP message (e.g. Dicover, Offer, Request, ACK).

(54) DHCP Server Identifier

example data:

```
    Length: 4
    DHCP Server Identifier: 10.0.1.21 (10.0.1.21)
```

Set by the server so clients can differentiate between different lease offers.

(55) Parameter Request List
example data:

```
    Length: 13
    Parameter Request List Item: (1) Subnet Mask
    Parameter Request List Item: (28) Broadcast Address
    Parameter Request List Item: (2) Time Offset
    Parameter Request List Item: (3) Router
    Parameter Request List Item: (15) Domain Name
    Parameter Request List Item: (6) Domain Name Server
    Parameter Request List Item: (119) Domain Search
    Parameter Request List Item: (12) Host Name
    Parameter Request List Item: (44) NetBIOS over TCP/IP Name Server
    Parameter Request List Item: (47) NetBIOS over TCP/IP Scope
    Parameter Request List Item: (26) Interface MTU
    Parameter Request List Item: (121) Classless Static Route
    Parameter Request List Item: (42) Network Time Protocol Servers
```

Set by the client to request values for specified configuration parameters.

(255) End
example data:

```
    Option End: 255
```

Marks the end of the options.

1. 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 `/etc/dhcpd.leases` at the DHCP server and `/var/lib/dhcp3/dhclient-eth0.lease` at the DHCP client (note that the latter name may differ).
   - To interpret the content of the files, refer to the manual pages of dhcpd.conf, dhcp-options, and dhcpd.leases.
   - Save the files that contain the information on current leases.
   - Observe how a DHCP client renews a lease and save the captured traffic to a file.
     - What type of DHCP message can be observed?
     - How long does a DHCP client wait until it attempts to renew its lease?
   - Stop the process that runs the DHCP server by terminating the process `dhcpd` with the command

     ```
     PC2% pkill dhcpd
     ```

Observe what the DHCP client does when it cannot reach the DHCP server. Use the command `ifconfig -a` to see how long the DHCP client waits until it releases the leased IP address.

- Restart the DHCP server process by typing

```
PC2% dhcpd
```

2. Starting more DHCP clients: Repeat the instructions in Step 2 and start DHCP clients on PC3 and PC4.

Question 2.C.4.a)

The expected outcome is that PC4 receives an IP address, but that PC3 is not successful. Why is the negative outcome for PC3 expected?

This happens because the DCHP relay is not configured yet. Without a DHCP relay, PC4 cannot receive and send DHCP packets.

Question 2.C.4.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?

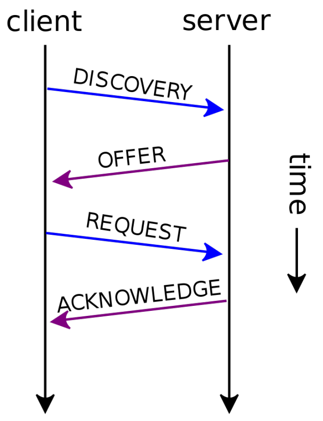
PC1: 10.0.1.2
PC4: 10.0.1.3

The assignment of IP addresses seems to be in incrementing order.

Question 2.C.a)

Use a figure to explain the packets that were exchanged by the DHCP client and the DHCP server as part of the process of acquiring an IP address.

We found this image on Wikipedia. For explanations, see Question 2.C.2.e).



Question 2.C.b)

Explain the entries in the lease file. How is the content of the lease file used when a DHCP server cannot contact the DHCP server?

interface: the interface the lease counts for
fixed-address: the IP address received by the server
option subnet-mask: the subnet mask received by the server
option routers: available DHCP servers
option dhcp-lease-time: total duration of the lease in seconds
option dhcp-message-type: type of DHCP message sent for this lease entry

option dhcp-server-identifier: the DHCP server that the client received this lease from
renew: timestamp of when this lease was received
rebind: timestamp of then to enter rebinding state
expire: timestamp of when this lease expires

The client can reach the server using the dhcp-server-identifier option. If the server cannot be reached by the rebind time, the client will enter rebinding state. From here on, it will try to contact any DCHP server by broadcasting a DHCP Discovery package.

Question 2.C.c)

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.

In de rfc staat

```
DHCP uses UDP as its transport protocol. DHCP messages from a client
 to a server are sent to the 'DHCP server' port (67), and DHCP
 messages from a server to a client are sent to the 'DHCP client' port
 (68). A server with multiple network address (e.g., a multi-homed
 host) MAY use any of its network addresses in outgoing DHCP messages.
```

????

Op SO:
http://superuser.com/questions/927849/why-does-dhcp-uses-udp-port-67-and-68-for-the-communication-between-the-client-a
http://serverfault.com/questions/517223/why-does-dhcp-have-fixed-client-and-server-port-numbers
http://stackoverflow.com/questions/1790960/why-dhcp-client-listens-on-port-68

Question 2.C.d)

Another protocol that can be used to assign IP addresses is the Reverse ARP (RARP) protocol. Compare the services provided by RARP and DHCP.

** todo **

Exercise 2-d: DHCP relay agent

⚠ *You will use Wireshark in this exercise. Do not forget to append the binary dump (pcap format) to your lab report*

A DHCP relay agent can forward DHCP packets when both 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 7.6

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:

```
Router> enable Password: <enable secret>
Router1# configure terminal
Router1(config)
Router1(config) ip forward-protocol udp
Router1(config) interface vlan1
Router1(config-if) ip helper-address 10.0.2.21
Router1(config-if) end
```

The following explains some of the above used commands:

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

`ip helper-address 10.0.2.21`
: *The DHCP request packets received on vlan1 will be forwarded to the DHCP server with address 10.0.2.21.*

2. Start Wireshark on PC2 and PC3.

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

4. Start a DHCP client on PC3 with

```
PC3% dhclient eth0
```

5. Verify that an IP address has been assigned to PC3. According to the configuration file, the DHCP configuration on network 10.0.2.0/24 does not set a default router. Verify that this is correct, by inspecting the routing table.

Question 2.D.a)

Include the Wireshark data of the first three DHCP packets that are exchanged between PC3 and PC2.

** todo: nakijken of dit juist is **

From "/Lab 7/traces/2.D.PC3.pcap":

```
    No.      Time          Source                    Destination           Protocol  Length
         Info
       3 3.409639     0.0.0.0                   255.255.255.255       DHCP      342
          DHCP Request  - Transaction ID 0x72f9f558

Frame 3: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr 11, 2016 15:19:03.482791000 CEST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1460380743.482791000 seconds
    [Time delta from previous captured frame: 1.410178000 seconds]
    [Time delta from previous displayed frame: 1.410178000 seconds]
    [Time since reference or first frame: 3.409639000 seconds]
    Frame Number: 3
    Frame Length: 342 bytes (2736 bits)
    Capture Length: 342 bytes (2736 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:udp:bootp]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
Ethernet II, Src: IntelCor_36:39:c7 (68:05:ca:36:39:c7), Dst: Broadcast (ff:ff:ff:
    ff:ff:ff)
    Destination: Broadcast (ff:ff:ff:ff:ff:ff)
        Address: Broadcast (ff:ff:ff:ff:ff:ff)
        .... ..1. .... .... .... .... = LG bit: Locally administered address (this
            is NOT the factory default)
        .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
    Source: IntelCor_36:39:c7 (68:05:ca:36:39:c7)
        Address: IntelCor_36:39:c7 (68:05:ca:36:39:c7)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IP (0x0800)
Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255
    (255.255.255.255)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x10 (DSCP 0x04: Unknown DSCP; ECN: 0x00: Not-
        ECT (Not ECN-Capable Transport))
        0001 00.. = Differentiated Services Codepoint: Unknown (0x04)
        .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable
            Transport) (0x00)
    Total Length: 328
    Identification: 0x0000 (0)
    Flags: 0x00
        0... .... = Reserved bit: Not set
        .0.. .... = Don't fragment: Not set
        ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 128
    Protocol: UDP (17)
    Header checksum: 0x3996 [validation disabled]
        [Good: False]
        [Bad: False]
    Source: 0.0.0.0 (0.0.0.0)
    Destination: 255.255.255.255 (255.255.255.255)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
    Source port: bootpc (68)
    Destination port: bootps (67)
    Length: 308
    Checksum: 0xfc20 [validation disabled]
        [Good Checksum: False]
        [Bad Checksum: False]
Bootstrap Protocol
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x72f9f558
    Seconds elapsed: 0
    Bootp flags: 0x0000 (Unicast)
        0... .... .... .... = Broadcast flag: Unicast
        .000 0000 0000 0000 = Reserved flags: 0x0000
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 0.0.0.0 (0.0.0.0)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: IntelCor_36:39:c7 (68:05:ca:36:39:c7)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
    Option: (53) DHCP Message Type
        Length: 1
        DHCP: Request (3)
    Option: (50) Requested IP Address
        Length: 4
        Requested IP Address: 10.0.3.2 (10.0.3.2)
    Option: (12) Host Name
```

```
        Length: 7
        Host Name: lab2pc1
    Option: (55) Parameter Request List
        Length: 13
        Parameter Request List Item: (1) Subnet Mask
        Parameter Request List Item: (28) Broadcast Address
        Parameter Request List Item: (2) Time Offset
        Parameter Request List Item: (3) Router
        Parameter Request List Item: (15) Domain Name
        Parameter Request List Item: (6) Domain Name Server
        Parameter Request List Item: (119) Domain Search
        Parameter Request List Item: (12) Host Name
        Parameter Request List Item: (44) NetBIOS over TCP/IP Name Server
        Parameter Request List Item: (47) NetBIOS over TCP/IP Scope
        Parameter Request List Item: (26) Interface MTU
        Parameter Request List Item: (121) Classless Static Route
        Parameter Request List Item: (42) Network Time Protocol Servers
    Option: (255) End
        Option End: 255
    Padding

No.     Time          Source                Destination           Protocol Length
    Info
   4 3.438875     10.0.3.1              10.0.3.2              DHCP     342
       DHCP ACK      - Transaction ID 0x72f9f558

Frame 4: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr 11, 2016 15:19:03.512027000 CEST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1460380743.512027000 seconds
    [Time delta from previous captured frame: 0.029236000 seconds]
    [Time delta from previous displayed frame: 0.029236000 seconds]
    [Time since reference or first frame: 3.438875000 seconds]
    Frame Number: 4
    Frame Length: 342 bytes (2736 bits)
    Capture Length: 342 bytes (2736 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:udp:bootp]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
Ethernet II, Src: Cisco_17:01:29 (00:0d:65:17:01:29), Dst: IntelCor_36:39:c7
    (68:05:ca:36:39:c7)
    Destination: IntelCor_36:39:c7 (68:05:ca:36:39:c7)
        Address: IntelCor_36:39:c7 (68:05:ca:36:39:c7)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: Cisco_17:01:29 (00:0d:65:17:01:29)
        Address: Cisco_17:01:29 (00:0d:65:17:01:29)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IP (0x0800)
Internet Protocol Version 4, Src: 10.0.3.1 (10.0.3.1), Dst: 10.0.3.2 (10.0.3.2)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (
        Not ECN-Capable Transport))
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable
            Transport) (0x00)
    Total Length: 328
    Identification: 0x0053 (83)
    Flags: 0x00
        0... .... = Reserved bit: Not set
        .0.. .... = Don't fragment: Not set
        ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 255
    Protocol: UDP (17)
    Header checksum: 0xa04f [validation disabled]
        [Good: False]
        [Bad: False]
    Source: 10.0.3.1 (10.0.3.1)
    Destination: 10.0.3.2 (10.0.3.2)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
    Source port: bootps (67)
    Destination port: bootpc (68)
    Length: 308
    Checksum: 0x6f1f [validation disabled]
        [Good Checksum: False]
        [Bad Checksum: False]
Bootstrap Protocol
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 1
    Transaction ID: 0x72f9f558
    Seconds elapsed: 0
    Bootp flags: 0x0000 (Unicast)
        0... .... .... .... = Broadcast flag: Unicast
        .000 0000 0000 0000 = Reserved flags: 0x0000
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 10.0.3.2 (10.0.3.2)
    Next server IP address: 10.0.2.21 (10.0.2.21)
    Relay agent IP address: 10.0.3.1 (10.0.3.1)
    Client MAC address: IntelCor_36:39:c7 (68:05:ca:36:39:c7)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
    Option: (53) DHCP Message Type
        Length: 1
        DHCP: ACK (5)
    Option: (54) DHCP Server Identifier
        Length: 4
        DHCP Server Identifier: 10.0.2.21 (10.0.2.21)
    Option: (51) IP Address Lease Time
        Length: 4
        IP Address Lease Time: (600s) 10 minutes
    Option: (1) Subnet Mask
        Length: 4
        Subnet Mask: 255.255.255.0 (255.255.255.0)
    Option: (3) Router
        Length: 4
        Router: 10.0.3.1 (10.0.3.1)
    Option: (255) End
        Option End: 255
    Padding

No.     Time           Source                Destination           Protocol Length
    Info
      5 3.439840       10.0.3.1              10.0.3.2              DHCP     342
          DHCP ACK       - Transaction ID 0x72f9f558

Frame 5: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr 11, 2016 15:19:03.512992000 CEST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1460380743.512992000 seconds
    [Time delta from previous captured frame: 0.000965000 seconds]
    [Time delta from previous displayed frame: 0.000965000 seconds]
    [Time since reference or first frame: 3.439840000 seconds]
    Frame Number: 5
    Frame Length: 342 bytes (2736 bits)
    Capture Length: 342 bytes (2736 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:udp:bootp]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
Ethernet II, Src: Cisco_17:01:29 (00:0d:65:17:01:29), Dst: IntelCor_36:39:c7
    (68:05:ca:36:39:c7)
    Destination: IntelCor_36:39:c7 (68:05:ca:36:39:c7)
        Address: IntelCor_36:39:c7 (68:05:ca:36:39:c7)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: Cisco_17:01:29 (00:0d:65:17:01:29)
        Address: Cisco_17:01:29 (00:0d:65:17:01:29)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IP (0x0800)
Internet Protocol Version 4, Src: 10.0.3.1 (10.0.3.1), Dst: 10.0.3.2 (10.0.3.2)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (
        Not ECN-Capable Transport))
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable
            Transport) (0x00)
    Total Length: 328
    Identification: 0x0054 (84)
    Flags: 0x00
        0... .... = Reserved bit: Not set
        .0.. .... = Don't fragment: Not set
        ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 255
    Protocol: UDP (17)
    Header checksum: 0xa04e [validation disabled]
        [Good: False]
        [Bad: False]
    Source: 10.0.3.1 (10.0.3.1)
    Destination: 10.0.3.2 (10.0.3.2)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
    Source port: bootps (67)
    Destination port: bootpc (68)
    Length: 308
    Checksum: 0x6f1f [validation disabled]
        [Good Checksum: False]
        [Bad Checksum: False]
Bootstrap Protocol
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 1
    Transaction ID: 0x72f9f558
    Seconds elapsed: 0
    Bootp flags: 0x0000 (Unicast)
        0... .... .... .... = Broadcast flag: Unicast
        .000 0000 0000 0000 = Reserved flags: 0x0000
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 10.0.3.2 (10.0.3.2)
```

```
Next server IP address: 10.0.2.21 (10.0.2.21)
Relay agent IP address: 10.0.3.1 (10.0.3.1)
Client MAC address: IntelCor_36:39:c7 (68:05:ca:36:39:c7)
Client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
Option: (53) DHCP Message Type
    Length: 1
    DHCP: ACK (5)
Option: (54) DHCP Server Identifier
    Length: 4
    DHCP Server Identifier: 10.0.2.21 (10.0.2.21)
Option: (51) IP Address Lease Time
    Length: 4
    IP Address Lease Time: (600s) 10 minutes
Option: (1) Subnet Mask
    Length: 4
    Subnet Mask: 255.255.255.0 (255.255.255.0)
Option: (3) Router
    Length: 4
    Router: 10.0.3.1 (10.0.3.1)
Option: (255) End
    Option End: 255
Padding
```

Question 2.D.6.a)

Does the DHCP relay server modify DHCP packets or the IP header? If so, what are the modifications?

In the IP header the Source and Destination addresses were changed. The relay server set the DHCP server as the destination and the address of the interface on which the DHCP client is connected as the source (vlan1 in this case).

No changes in the DHCP part **part? hoe zeg je het juist** were observed.

Question 2.D.6.b)

How does the relay agent redirect the replies from the DHCP server? Does it LAB 7- PAGE 20 broadcast them or unicast them to the DHCP client?

The relay agent unicasts the replies from the DHCP server to the DHCP client, just like the DHCP server did in the previous excercise.

** kan iemand confirmen? **

Question 2.D.6.c)

Is there a difference in the response of the DHCP server as compared to the DHCP configuration of PC1? If so, explain the difference.

** wat ik denk dat het moet zijn **

There is no difference. The DHCP client does not know that there is no real DHCP server on its subnet.

Question 2.D.6.d)

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

** nakijken met nieuwe trace files ** The source IP address of the DHCP packets has Router1's vlan address. This address is also filled in in "Relay agent IP address" in the Bootstrap Protocol part of the packet. The server can know on which network PC3 is located by looking at this address.

Question 2.D.6.e)

What is the destination IP address of the first DHCP packet that the DHCP server sends to

PC3?

It is the IP address of Router1's vlan1 IP address. Router1 will then forward this packet to PC3. It knows how to reach PC3 because of the "Client MAC address" value in the Bootstrap Protocol part of the packet.

Question 2.D.c)

What happens if a network has multiple DHCP servers?

This depends on the configuration of the DHCP client. It can compare offers from different DHCP servers and choose the best one, or simply work with the server that answerred first and ignore the rest. ** todo/meer info, ben niet zeker **

7.3 Combining NAT and DHCP

Figure 7.3 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 âĂIJpublicâĂİ 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.

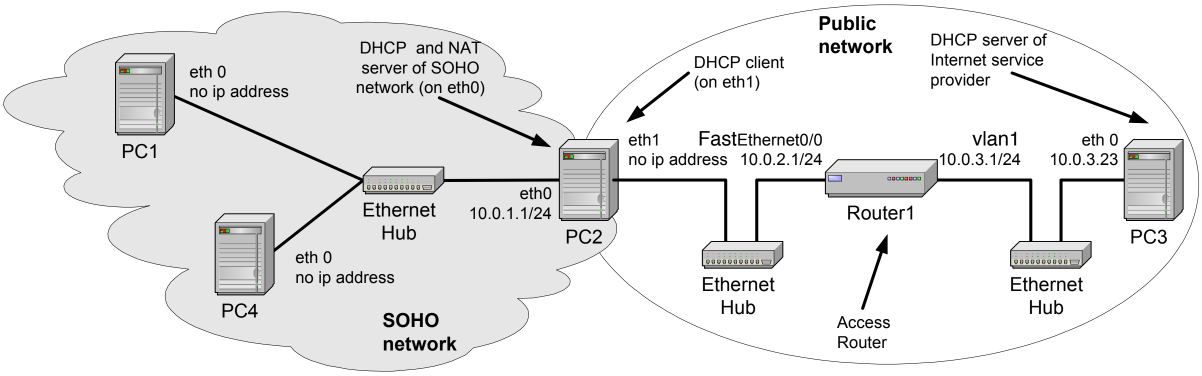


Figure 7.3: Network configuration for Part 3.

Part 1. Exercise 3:

You will use Wireshark in this exercise. Do not forget to append the binary dump (pcap format) to your lab report

The network configuration is shown as Figure 7.3. (The connections of the cables are identical to Figure 7.2). To reset the configuration of all machines, we recommend rebooting the PCs and the router.

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

- Configure PC3 with IP address 10.0.3.23/24 on interface *eth0* and with default gateway 10.0.3.1.
- Configure and 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, and addresses in the range 10.0.3.2-10.0.3.10 are assigned for requests on network 10.0.3.0/24.

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 *FastEthernet0/0* and 10.0.3.1/24 on interface *vlan1*.
   - The routing table of Router1 should reflect that all traffic to network 10.0.2.0/24 is sent on interface *FastEthernet0/0*, and all other traffic is sent on interface *vlan1*.
   - Configure Router1 as a DHCP relay agent, so that requests from DHCP client PC2 reach DHCP server PC3.

3. SOHO Router: PC2 is the SOHO router.

   - Set up PC2 so that it is a DHCP client on interface *eth1*.
   - Set up PC2 as an IP router. That is, IP forwarding must be enabled. The routing table entries must reflect that traffic to network 10.0.1.0/24 must be routed on interface *eth0*, and all other traffic must be sent to Router1 at 10.0.2.1.
   - Configure PC2 as DHCP server on interface *eth0* for addresses in the range 0.0.1.2 - 10.0.1.10. Execute the following command to start a DHCP server process on PC2:

     ```
     PC2% dhcpd eth0
     ```

   - 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:

     ```
     iptables -t nat -A POSTROUTING -j MASQUERADE -o eth1 -s 10.0.1.0/24
     ```

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

   - Set up PC1 and PC4 as DHCP clients on interfaces *eth0*.

5. Collecting the results:

   - Display the routing tables from all PCs with `netstat -rn`, and the IP configuration with `ifconfig -a`, and save the results.
   - What are the IP addresses assigned to PC1 and PC4? How are the IP addresses mapped to the public IP address defined on the NAT server PC2?
   - Display and save the NAT table of PC2.
   - Start Wireshark on PC1 (*eth0*), PC2 (*eth1*), and PC3 (*eth0*).
   - Issue a `ping` command from PC1 to PC3:

     ```
     PC1% ping -c 5 10.0.3.23
     ```

   - Save the traffic captured by Wireshark on one of the PCs to a file.

Question 3.a)

Include the Wireshark data from the first ICMP Request and ICMP Reply messages.
From "/Lab 7/traces/3-5.6.PC1.pcap":

```
No.     Time           Source                Destination           Protocol Length
    Info
      1 0.000000       10.0.1.2              10.0.3.23             ICMP     98
         Echo (ping) request  id=0x226e, seq=1/256, ttl=64 (reply in 2)

Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr 11, 2016 16:42:32.134218000 CEST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1460385752.134218000 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 98 bytes (784 bits)
    Capture Length: 98 bytes (784 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:icmp:data]
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
Ethernet II, Src: IntelCor_36:33:a0 (68:05:ca:36:33:a0), Dst: IntelCor_36:31:f0
    (68:05:ca:36:31:f0)
    Destination: IntelCor_36:31:f0 (68:05:ca:36:31:f0)
        Address: IntelCor_36:31:f0 (68:05:ca:36:31:f0)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: IntelCor_36:33:a0 (68:05:ca:36:33:a0)
        Address: IntelCor_36:33:a0 (68:05:ca:36:33:a0)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IP (0x0800)
Internet Protocol Version 4, Src: 10.0.1.2 (10.0.1.2), Dst: 10.0.3.23 (10.0.3.23)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (
        Not ECN-Capable Transport))
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable
            Transport) (0x00)
    Total Length: 84
    Identification: 0x6925 (26917)
    Flags: 0x02 (Don't Fragment)
        0... .... = Reserved bit: Not set
        .1.. .... = Don't fragment: Set
        ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (1)
    Header checksum: 0xb96b [validation disabled]
        [Good: False]
        [Bad: False]
    Source: 10.0.1.2 (10.0.1.2)
    Destination: 10.0.3.23 (10.0.3.23)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xfea2 [correct]
    Identifier (BE): 8814 (0x226e)
    Identifier (LE): 28194 (0x6e22)
    Sequence number (BE): 1 (0x0001)
    Sequence number (LE): 256 (0x0100)
    [Response frame: 2]
    Timestamp from icmp data: Apr 11, 2016 16:42:32.000000000 CEST
    [Timestamp from icmp data (relative): 0.134218000 seconds]
    Data (48 bytes)

0000  32 0c 02 00 00 00 00 00 10 11 12 13 14 15 16 17   2...............
0010  18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27   ........ !"#$%&'
0020  28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37   ()*+,-./01234567
        Data: 320c020000000000101112131415161718191a1b1c1d1e1f...
        [Length: 48]

No.     Time           Source                Destination           Protocol Length
    Info
      2 0.001862       10.0.3.23             10.0.1.2              ICMP     98
          Echo (ping) reply    id=0x226e, seq=1/256, ttl=62 (request in 1)

Frame 2: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr 11, 2016 16:42:32.136080000 CEST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1460385752.136080000 seconds
    [Time delta from previous captured frame: 0.001862000 seconds]
    [Time delta from previous displayed frame: 0.001862000 seconds]
    [Time since reference or first frame: 0.001862000 seconds]
    Frame Number: 2
    Frame Length: 98 bytes (784 bits)
    Capture Length: 98 bytes (784 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:icmp:data]
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
Ethernet II, Src: IntelCor_36:31:f0 (68:05:ca:36:31:f0), Dst: IntelCor_36:33:a0
    (68:05:ca:36:33:a0)
    Destination: IntelCor_36:33:a0 (68:05:ca:36:33:a0)
        Address: IntelCor_36:33:a0 (68:05:ca:36:33:a0)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: IntelCor_36:31:f0 (68:05:ca:36:31:f0)
        Address: IntelCor_36:31:f0 (68:05:ca:36:31:f0)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory
            default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IP (0x0800)
Internet Protocol Version 4, Src: 10.0.3.23 (10.0.3.23), Dst: 10.0.1.2 (10.0.1.2)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (
        Not ECN-Capable Transport))
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable
            Transport) (0x00)
    Total Length: 84
    Identification: 0xeeaa (61098)
    Flags: 0x00
        0... .... = Reserved bit: Not set
        .0.. .... = Don't fragment: Not set
        ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 62
    Protocol: ICMP (1)
    Header checksum: 0x75e6 [validation disabled]
        [Good: False]
        [Bad: False]
    Source: 10.0.3.23 (10.0.3.23)
    Destination: 10.0.1.2 (10.0.1.2)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x06a3 [correct]
    Identifier (BE): 8814 (0x226e)
    Identifier (LE): 28194 (0x6e22)
    Sequence number (BE): 1 (0x0001)
    Sequence number (LE): 256 (0x0100)
    [Request frame: 1]
    [Response time: 1,862 ms]
    Timestamp from icmp data: Apr 11, 2016 16:42:32.000000000 CEST
    [Timestamp from icmp data (relative): 0.136080000 seconds]
    Data (48 bytes)

0000  32 0c 02 00 00 00 00 00 10 11 12 13 14 15 16 17   2...............
0010  18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27   ........ !"#$%&'
0020  28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37   ()*+,-./01234567
        Data: 320c020000000000101112131415161718191a1b1c1d1e1f...
        [Length: 48]
```

Question 3.b)

Include the routing table and the output of the `ifconfig` command from all PCs.

PC1:

```
student@lab2pc1:~$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.0.1.1        0.0.0.0         UG        0 0          0 eth0
10.0.1.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0



student@lab2pc1:~$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 68:05:ca:36:33:a0
          inet addr:10.0.1.2  Bcast:10.0.1.255  Mask:255.255.255.0
          inet6 addr: fe80::6a05:caff:fe36:33a0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:783 errors:0 dropped:0 overruns:0 frame:0
          TX packets:830 errors:0 dropped:0 overruns:0 carrier:0
          collisions:12 txqueuelen:1000
          RX bytes:140872 (140.8 KB)  TX bytes:150460 (150.4 KB)
          Interrupt:19 Memory:f06c0000-f06e0000

eth1      Link encap:Ethernet  HWaddr 68:05:ca:39:cc:79
          inet addr:10.0.5.11  Bcast:10.0.5.255  Mask:255.255.255.0
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:46 errors:0 dropped:0 overruns:0 frame:0
          TX packets:58 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7051 (7.0 KB)  TX bytes:8069 (8.0 KB)
          Interrupt:16 Memory:f05c0000-f05e0000

internet  Link encap:Ethernet  HWaddr d0:50:99:55:a9:42
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:224985 errors:0 dropped:0 overruns:0 frame:0
          TX packets:144710 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:109926155 (109.9 MB)  TX bytes:14609737 (14.6 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:4561 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4561 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:902418 (902.4 KB)  TX bytes:902418 (902.4 KB)
```

PC2:

```
student@lab2pc1:~$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.0.2.1        0.0.0.0         UG        0 0          0 eth0
10.0.1.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 eth1



student@lab2pc1:~$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 68:05:ca:36:31:f0
          inet addr:10.0.1.1  Bcast:10.0.1.255  Mask:255.255.255.0
          inet6 addr: fe80::6a05:caff:fe36:31f0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:989 errors:0 dropped:0 overruns:0 frame:0
          TX packets:665 errors:0 dropped:0 overruns:0 carrier:0
          collisions:9 txqueuelen:1000
          RX bytes:172049 (172.0 KB)  TX bytes:103391 (103.3 KB)
          Interrupt:19 Memory:f06c0000-f06e0000

eth1      Link encap:Ethernet  HWaddr 68:05:ca:39:e1:36
          inet addr:10.0.2.2  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::6a05:caff:fe39:e136/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1563 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1600 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:162818 (162.8 KB)  TX bytes:205981 (205.9 KB)
          Interrupt:16 Memory:f05c0000-f05e0000

internet  Link encap:Ethernet  HWaddr d0:50:99:55:a9:10
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:103555 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8705 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:117730597 (117.7 MB)  TX bytes:1148488 (1.1 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:5361 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5361 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:901868 (901.8 KB)  TX bytes:901868 (901.8 KB)
```

PC3:

```
student@lab2pc1:~$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.0.3.1        0.0.0.0         UG        0 0          0 eth0
10.0.3.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0



student@lab2pc1:~$ ifconfig -a
```

```
eth0      Link encap:Ethernet  HWaddr 68:05:ca:36:39:c7
          inet addr:10.0.3.23  Bcast:10.0.3.255  Mask:255.255.255.0
          inet6 addr: fe80::6a05:caff:fe36:39c7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:399 errors:0 dropped:0 overruns:0 frame:0
          TX packets:265 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:33766 (33.7 KB)  TX bytes:47715 (47.7 KB)
          Interrupt:19 Memory:f06c0000-f06e0000

eth2      Link encap:Ethernet  HWaddr 68:05:ca:36:51:3f
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:16 Memory:f05c0000-f05e0000

internet  Link encap:Ethernet  HWaddr d0:50:99:55:a9:28
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:247 errors:0 dropped:0 overruns:0 frame:0
          TX packets:247 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:23782 (23.7 KB)  TX bytes:23782 (23.7 KB)
```

PC4:

```
root@lab2pc1:~/labdata# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.0.1.1        0.0.0.0         UG        0 0          0 eth0
10.0.1.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0



root@lab2pc1:~/labdata# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 68:05:ca:39:e1:2f
          inet addr:10.0.1.5  Bcast:10.0.1.255  Mask:255.255.255.0
          inet6 addr: fe80::6a05:caff:fe39:e12f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:702 errors:0 dropped:0 overruns:0 frame:0
          TX packets:572 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:154200 (154.2 KB)  TX bytes:75589 (75.5 KB)
          Interrupt:19 Memory:f06c0000-f06e0000

eth1      Link encap:Ethernet  HWaddr 68:05:ca:39:e1:32
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:97 errors:0 dropped:0 overruns:0 frame:0
          TX packets:85 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6918 (6.9 KB)  TX bytes:19392 (19.3 KB)
          Interrupt:16 Memory:f05c0000-f05e0000

eth1:avahi Link encap:Ethernet  HWaddr 68:05:ca:39:e1:32
          inet addr:169.254.7.195  Bcast:169.254.255.255  Mask:255.255.0.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          Interrupt:16 Memory:f05c0000-f05e0000

internet  Link encap:Ethernet  HWaddr d0:50:99:55:a9:0c
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:34701 errors:0 dropped:0 overruns:0 frame:0
          TX packets:20487 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:27866506 (27.8 MB)  TX bytes:3327141 (3.3 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:3035 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3035 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:415396 (415.3 KB)  TX bytes:415396 (415.3 KB)
```

Question 3.c)

Include the NAT table form PC2.

```
student@lab2pc1:~$ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE  all  --  10.0.1.0/24          anywhere
```