

Name 1:

Name 2:

COMPUTER NETWORKING

LAB EXERCISES (TP) 2

L1 v.s. L2 v.s. L3, NAT AND TROUBLESHOOTING

October 8, 2015

Abstract

In this Lab you will work with the virtual environment introduced in Lab 1. First you will see the different behavior of networking devices that work on Layer 1, layer 2 and layer 3; then you will configure your virtual network to be able to access the Internet; later you will help Anakin and Padme to fix their networking problems when a common enemy, Sidious, changes the configuration in their network. Finally you will practice reading TCP-IP headers to collect and correlate information.

1 PREPARING THE LAB

1.1 LAB REPORT

Type your answers in this document. We recommend you use Adobe Reader XI to open this PDF. When you finish, save the report and upload it on moodle. Don't forget to write your names on the first page of the report. **The deadline is Wednesday, October 21th, 23:59:59**

1.2 SETTING UP VIRTUAL MACHINES

In this Lab, you will work with the four virtual machines that you created in Lab 1. Check the interface naming and delete `70-persistent-net.rules` file in case you need it. We make the same interface labeling `eth0`, `eth1` and `eth2`

1.3 IPV4 CONNECTIVITY PROBLEMS IN GNS3

If you have problems pinging between interfaces of PCs that are directly connected (on-link), check the workaround posted in moodle under important announcements: <http://moodle.epfl.ch/mod/forum/discuss.php?d=646>

2 LAYER 1 VS. LAYER 2 VS. LAYER 3 NETWORKING

The aim of this section is to illustrate the difference between networking devices that work at layer 1, layer 2 and layer 3. For this exercise we only consider IPv4 addressing, and we will be using the same private IPv4 address space $10.10.0.0/16$, as in Lab 1.

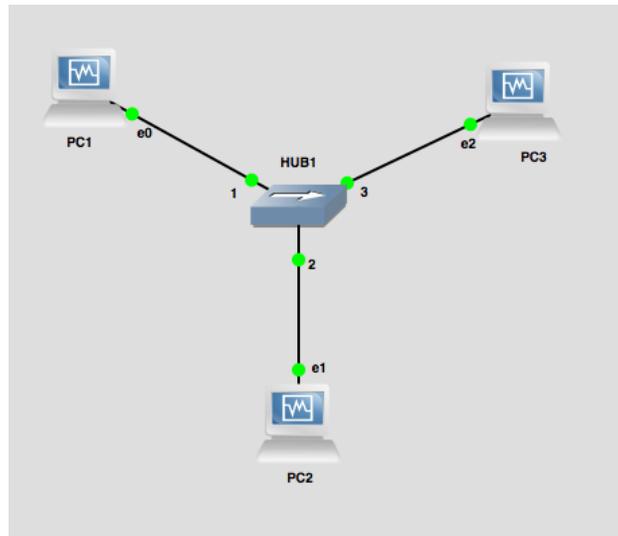


Figure 1: Network configuration in a star topology with a hub in the center

2.1 USING HUB AS A NETWORKING DEVICE

A hub is layer-one intermediate system that repeats bits (symbols) received from one of its ports to all other ports. In this section we analyze how it works.

Wire PC1, PC2 and PC3 as described in Figure 1. Use the subnet $10.10.14.0/24$ to provide IP addresses for the LAN and use the fourth byte to identify the PC: Use 1 for PC1 and 2 for PC2. For PC3 use the IP address $10.10.34.3/24$.

Start Wireshark on all three machines. From PC2, ping PC1.

```
# ping 10.10.14.1
```



Q1/ Is there any difference between the traffic captured by the three PCs?

[A1]

Q2/ Explain why do you see these results

[A2]

Q3/ Ping from PC2 to PC3. Did you see *echo request* packets?. Explain why you are not able to ping PC3 from PC2 based on your findings

[A3]

2.2 USING A BRIDGE AS A NETWORKING DEVICE

A bridge is a link-layer intermediate system which expands a LAN by making forwarding decision based on destination MAC-address. In this section you will learn how they work.

In GNS3, remove HUB1 and replace it with PC4 as depicted in Figure 2. Instead of PC4 we could use an Ethernet bridge from the pool of GNS3 devices, but in order to save time, we'll use the versatility of Linux and create an Ethernet bridge using PC4 and the `brctl` utility. `brctl` is a tool used to set up, maintain, and inspect the Ethernet bridge configuration in the Linux kernel.

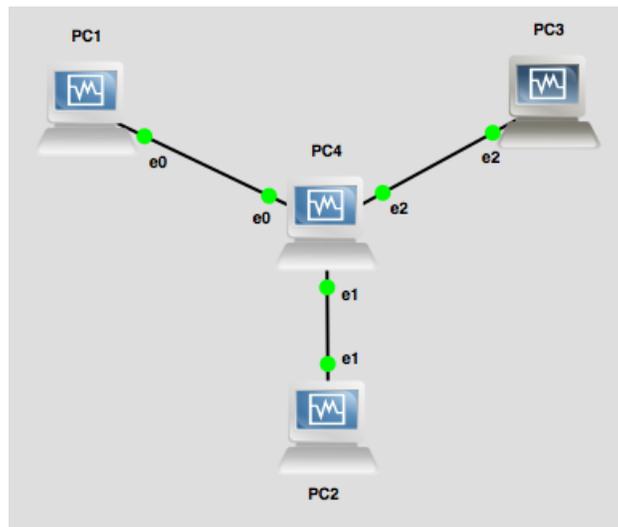


Figure 2: Network configuration in a star topology with PC4 in the center

The steps we will follow are:

1. Create a new Ethernet bridge instance on PC4
2. Add physical interfaces to the bridge
3. Remove any IPv4 addressing

Before configuring the bridge, let's make sure that all three interfaces `eth0`, `eth1` and `eth2` on PC4 are up. If an interface is down, it cannot be added to the bridge interface:

```
# ip link set eth0 up
# ip link set eth1 up
# ip link set eth2 up
```

Now, create a new bridge interface called `br0`, using:

```
# brctl addbr br0
```



Q4/ Write down the command you need to add the three interfaces of PC4 to the created bridge `br0`. Hint: you can check the how-to page for the `brctl` tool located in http://linuxcommand.org/man_pages/brctl8.html.

[A4]

Finally, we need to set the IP addresses of all PC4's interfaces to `0.0.0.0`:

```
# ifconfig eth0 0.0.0.0
# ifconfig eth1 0.0.0.0
# ifconfig eth2 0.0.0.0
```



Q5/ Why do we zero the IP addresses of PC4?

[A5]

Activate the bridge interface:

```
# ifconfig br0 up
```

The bridge is now configured!

Now, let's test our bridge configuration. Leave the same ip addressing scheme as in last section. Start a new capture in Wireshark in all PCs and monitor all interesting interfaces, From PC2, ping again PC1.

```
# ping 10.10.14.1
```



Q6/ Describe the different types of packets observed on PC1, PC2 and PC3

[A6]

Q7/ Explain the results. What is the difference compared to a hub?

[A7]

Now, focus on interfaces `eth0` and `eth1` of PC4. Ping again from PC2 to PC1.



Q8/ Write down your findings about the input/output traffic between the network interfaces `eth0` and `eth1`, specifically at source/destination MAC-addresses

[A8]

2.3 USING A PC AS A ROUTING DEVICE

We have already configured a router in Lab 1, but we did not address how it worked. In this section we learn about the process of routing a packet.

Let's start by enabling routing in PC4. The steps to fulfill this task are:

1. Remove the bridge configuration from PC4
2. Enable IP forwarding
3. Configure IP addresses
4. Set default gateways in PC1, PC2 and PC3.

First bring down the bridge interface. Use the same command syntax that you used for bringing it up.



Q9/ Type in the command you used for bringing down the bridge interface

[A9]

Now, delete the bridge interface using the following command:

```
# brctl delbr br0
```

Our next step is to enable IPv4 forwarding on PC4, just like we did in Lab1



Q10/ Type in the command you used to enable IPv4 forwarding on PC4

[A10]

Next step is to provide IP addresses to all the LANs. We will use the same IP addressing scheme as in Lab1 (private IPv4 address space `10.10.0.0/16`.)

- For the local network connecting PC4 to PC1 (LAN1), we will use the IPv4 network address prefix `10.10.14.0/24`. The host identifier (the fourth byte) should be 1 to indicate the PC1 and 4 for PC4.

- For the local network connecting PC4 to PC2 (LAN2), we will use the IPv4 network address prefix 10.10.24.0/24. The host identifier (the fourth byte) should be 2 to indicate the PC2 and 4 for PC4.
- For the local network connecting PC4 to PC3 (LAN3), we will use the IPv4 network address prefix 10.10.34.0/24. The host identifier (the fourth byte) should be 3 to indicate the PC3 and 4 for PC4.

The final step is to set the default gateways for PC1, PC2 and PC3 to its corresponding PC4's interface that is in the same LAN.



Q11/ Write down the commands used to set all default gateways

[A11]

It's time to learn about routing between two interfaces. Monitor the traffic in Wireshark of all PCs. From PC2, ping one more time PC1.

```
# ping 10.10.14.1
```



Q12/ Describe the different types of packets that you see in PC1, PC2 and PC3

[A12]

Now, focus again on interfaces `eth0` and `eth1` of PC4. Ping once more from PC2 to PC1.



Q13/ What changes are done to IP packets when they are routed between PC4's `eth0` and `eth1`?

[A13]

Q14/ What is the purpose of such changes?

[A14]

2.4 ROUTING WITH MULTIPLE HOPS

In this section we want to see what happens when we introduce a second intermediate routing device to the network. Let's start by setting the default gateway of PC4 to PC2's IP address, then on PC2 remove the default gateway and enable IPv4 forwarding.



Q15/ Type in the commands you need to do this

[A15]

Monitor `eth1` of PC2 and PC4. Try pinging from PC1 and PC3 to PC2.



Q16/ Based on Wireshark captures, explain why it does not work.

[A16]



Q17/ Write down **one single command** that fixes the problem (namely pinging from PC1 and PC3 to PC2), and specify the PC where you need to apply it

[A17]

Ping again from PC1 and PC3, and confirm that your fix solves the problem before moving to the next section.

3 CONNECTING VIRTUAL ENVIRONMENT TO THE REAL WORLD USING NETWORK ADDRESS TRANSLATION (NAT)

In this section we will use what we learned from Lab1 about manipulating the `iptables` filter. The purpose of the section is to connect the isolated virtual network that we have deployed so far, to the real Internet.

We will work in the network described in Figure 3. PC1 and PC3 are workstations, PC4 is an aggregation router, and PC2 a the perimeter router where we will have our connection to the real world. Note that the cloud is just illustrating that PC2 will be connected to the Internet through `eth0`, **there should be no cloud in the GNS3 configuration.**



Q18/ We have one real connection (IP address) to the real world, but we have two clients (PC1 and PC3) that require access to the internet. Which solution would you use to tackle this problem, and explain how would it solve the problem

[A18]

There are two main steps to connect your virtual environment to the real Internet:

1. We require a real IP address on `eth0` interface of PC2.
2. We need to masquerade the traffic coming from PC1 and PC3.

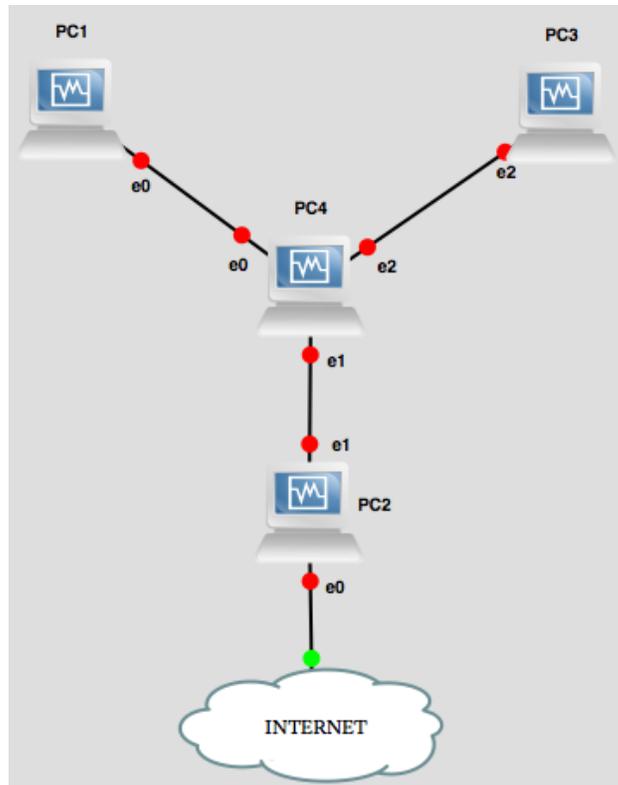


Figure 3: Network configuration with a connection to the real world

3.1 BORROWING AN IPV4 ADDRESS

As we said before, we need an *existing* (real) IPv4 address that can be used to connect to the Internet. The purpose of this section is to obtain such valid IPv4 address. There are three steps for that:

1. Create a bridge between a real interface in your host machine, and PC2's `eth0`.
2. Finding a suitable IPv4 address.
3. Setting up the IPv4 address of `eth0` on PC2.

3.1.1 BRIDGE BETWEEN THE PHYSICAL AND THE VIRTUAL INTERFACE

For the first step, the process is shown schematically in Figure 4, and we will cover it step by step.

To perform the bridging between physical and virtual network-adapters, perform the following configuration changes in the VirtualBox window. **Note: you do not require to stop GNS3 simulation:**

1. Select PC2, right-click and press the `Settings` option.
2. Go to the `Network` tab, and confirm that `Adapters 1` and `2` are enabled.
3. Confirm that `Adapter 2` is attached to “`Generic Driver`”
4. Attach `Adapter 1` to `Bridged Adapter`
5. Notice the `Name` of your physical network-adapter: if you have different network adapters on your host machine, you can choose anyone (that has an active Internet connection) to bridge. Typically, you can choose between the `Wireless adapter` and the `Ethernet (LAN) adapter` of your host machine. From now on, the chosen adapter will be just called the *physical adapter*.
6. Under the `Advanced options`, **make sure that `Cable connected` is checked**. Additionally, change

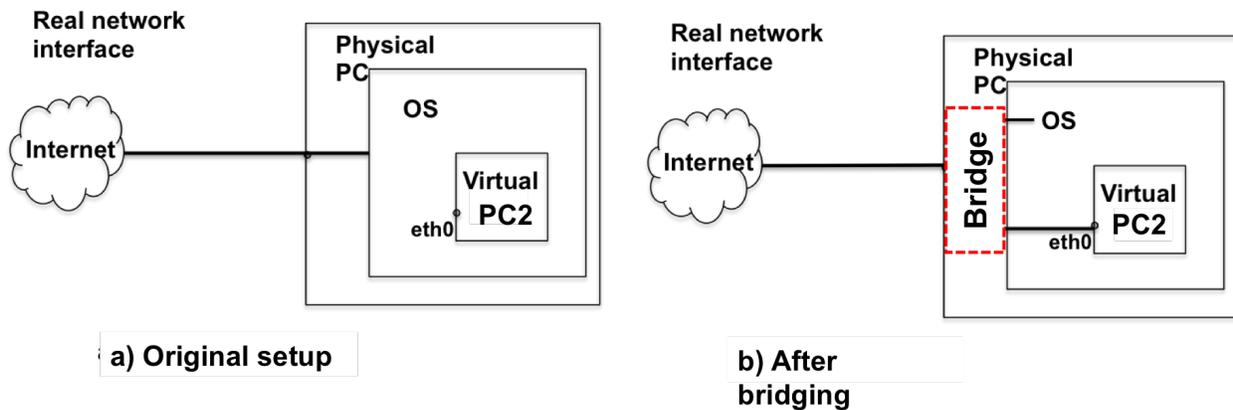


Figure 4: Bridging the network adapter

the `Promiscuous Mode` to “Allow All”. This option will forward any traffic captured by your physical adapter to your virtual network-adapter.

7. Press OK.

We have configured the bridge now. In order to test it we need to configure an IP address in PC2’s `eth0`.

3.1.2 CHOOSING THE BEST IP ADDRESS FOR PC2’S `ETH0`

The task in this section is to find a suitable IP address for PC2’s `eth0`. Run `ifconfig` and `route -n get default` (in Linux or Mac) or `ipconfig /all` (in Windows) and check your physical IP address, subnet mask and default gateway.



Q19/ Write down your physical IP address, subnet mask and default gateway. According to your network configuration, what is the range of IP addresses that we could use for the virtual adapter ?

[A19]

Q20/ How could you find the current non-used IP addresses in your LAN?. Is it safe to take any of them?

[A20]



Q21/ If you are doing this exercise outside EPFL (e.g. at your home), you will most likely get a private IP address and default gateway (e.g. `192.168.x.x` or `10.x.x.x`). Can you use these private IP addresses on PC2’s `eth0` to do NAT?. Explain why

[A21]

3.1.3 WIRING AND CONFIGURING `eth0` OF PC2

Before configuring your virtual-adapter, remove all IP address configuration from the physical adapter of your host machine. We will configure PC2's `eth0` with the same IP address as your physical adapter. The reason is because most of the recent Ethernet switches and wireless access points have MAC-layer security mechanisms that prevents a single MAC-address from having several IPv4 addresses.

Configure PC2's `eth0` according to your own physical-adapter's settings. Then use the command-line terminal to configure the file `/etc/resolv.conf` in order to set the DNS server `8.8.8.8` on PC2



Q22/ Type in the command you used to do it

[A22]

Test connectivity by pinging Google using the IP address and hostname.

3.2 NAT CONFIGURATION

We worked with the command used to configure NAT in Lab1, `iptables -t nat`, which manages the table that contains rules regarding address translations. In this section, we will analyze how does NAT work for different types of packets.

First let's see what happens when PC1/PC3 access to the Internet with their native IP address. Monitor with Wireshark the interface `eth0` of PC2. From PC1/PC3, ping to Google and its IP address.

```
# ping -c 5 www.google.com
```

```
# ping -c 5 173.194.40.41
```



Q23/ Analyze the packets coming from PC1/PC3 and explain why you are unable to reach Google

[A23]



Q24/ Propose the `iptables -t nat` command you need to properly configure NAT in PC2.

[A24]

Test from PC1 and PC3 and you have Internet connectivity by pinging Google and test that you have successfully configured your router to do NAT.

Next, let's explore how NAT works!!.

Do `tracert` to Google from PC3 and then from PC2, while capturing `eth0` and `eth1` traffic on PC2 using Wireshark. Explore the difference in the traffic on both cases.



Q25/ When doing `tracert` from PC3, what is the difference in the packets captured on PC2's `eth0` and `eth1`?

[A25]



Q26/ Focus on the `tracert` from PC2. What is the difference in the packets as compared to PC3

[A26]



Q27/ Which field in the UDP packet is used to identify the (local) source IP address of PC3 in order to properly forward incoming ICMP replies back to it?

[A27]

Do `ping` to Google from PC1 and PC2, while capturing the traffic on PC2 (both on `eth0` and `eth1`) using Wireshark. Explore the difference in the traffic in both cases.



Q28/ What is the difference in the request ICMP packets captured on `eth0` between packets sent from PC1 and packets sent from PC2?

[A28]



Q29/ Conclude how the incoming ICMP replies are forwarded back to PC1 when doing `ping` from PC1. In particular, which field in the request/reply ICMP packets was used to identify the (local) source IP address?

[A29]

4 TROUBLESHOOTING

As the title for this section suggests, in networking things will not always work out as expected. Before starting to work in this section, you will have to execute a script. This script will put a number of PCs into a problematic situation where something doesn't work. Your task is to find out what the problem is and propose a solution.

You should not perform any debug command or Wireshark capture in PC2. Assume PC2 is a router controlled by your Internet service provider (ISP) and you don't have access to it.

4.1 ABOUT GRADING THIS SECTION

The points given for your answer mostly depend on your explanations about how you located the problem!, so describe precisely your steps to locate and diagnose the fault. You should use the scientific method when answering the problem. The methodology you should use is the following:

1. Pose a hypothesis
2. Run experiments to validate the hypothesis
3. If validation is OK exit, else loop (go back to 1. by posing another hypothesis)

In your answer you should write down all steps. Specially, you should also write down all hypotheses that later proved to be wrong. We want to see the path you took to reach your final conclusion!

More specifically for this Lab: What were the commands you executed to get there? Up to which point did the system work as expected? What were the actions that never got executed but were expected? What was the packet that did not reach its destination? Where and why did it get dropped/lost?

4.2 WE WERE HACKED!!!

Anakin and Padme are roommates and close friends. Both of them are connected to the Internet through the same home router thus the same ISP. The configuration is the same as described in Figure 5, where PC1 is Padme's computer and PC3 is Anakin's computer, PC4 is the home router, shared between Anakin and Padme, and PC2 is the ISP's router.

Anakin and Padme are foreign students at EPFL and they use Facebook a lot to communicate with their relatives in their home town. None of them is an expert in computers and networking. They have a common enemy, Sidious, who plays them jokes and is a Computer Science student and network expert. Sidious told Padme and Anakin that he hacked their computers (PC1 and PC3) as well their home router (PC4) and that they would never navigate through Facebook again.

To simulate Sidious's malicious attack, we prepared a script for PC1, PC3 and PC4. For each PCx, you should download the script from the course web page on moodle. Unzip and extract the corresponding file for each PC:

```
# tar -xzvf lab2-script-pcX.tar.gz
```

Where x=1, 3, 4. This will create a directory `lab2-script-pcX/` containing the script used in this section for the selected PC. We recommend you to extract all scripts to your shared folder between host and virtual machines.

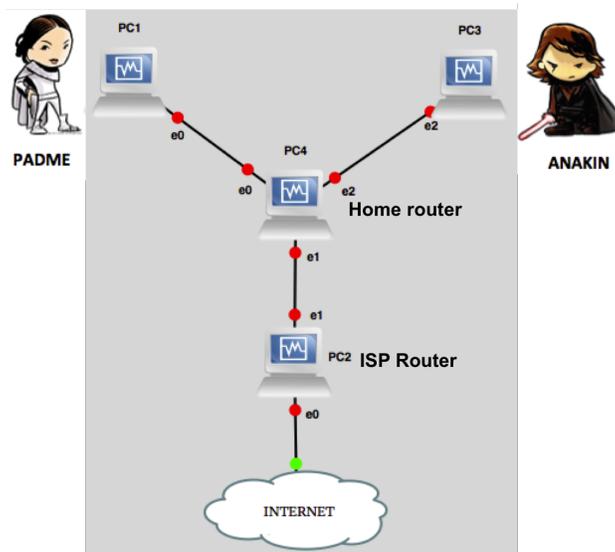


Figure 5: Troubleshooting configuration

Let's use PC1 as an example. Check that on PC1 you have a `lab2-script-pc1` directory with the following content:

```
# ls lab2-script-pc1
lab2_pb_pc1_x
```

Navigate to the `lab2-script-pc1` folder and execute the script. Make sure you **run the script as a superuser in Linux**. Make sure you repeat the same procedure for all PCs.

```
# cd lab2-script-pc1
# sudo ./lab2_pb_pc1_x
```

Note that if you stop the simulation in GNS3, then you need to reload the script on each of the PCs.

Now, your mission is to help Padme and Anakin to find out why they cannot navigate to Facebook anymore. You are only allowed to have Wireshark captures and use debug commands in PC1, PC3 and PC4. **You are not allowed to touch PC2.**



Q30/ Let's start with Padme. Open a browser on Padme's PC (PC1) and try to navigate to Facebook. Are there any problems in Padme's PC (PC1)? Enumerate them (if any) and write down how you find them and how you fix them.

[A30]



Q31/ It is the time for Anakin. Open a browser on Anakin's PC (PC3) and try to navigate to Facebook. Are there any problems in Anakin's PC (PC3)? Enumerate them (if any) and write down how you find them and how you fix them.

[A31]



Q32/ Is there any other problem in the network? Where is it most likely located (PC2 or PC4)? Write down how you find the problem

[A32]

5 READING IPV4 AND IPV6 PACKET HEADERS

In this section, you will observe IPv4 and IPv6 connections in different contexts. Your task is to answer the questions based only on the traces presented for each connection.

5.1 IPV4 PACKET HEADERS

Anakin is sitting in front of `lrcpc3` workstation and connects to `'ezinfo.ethz.ch'`. An ethical hacker has read all the frames passing on the network. Here are two packets resulting from this activity:

```
ETHER: ----- Ether Header -----
ETHER:
ETHER: Packet 2 arrived at 19:03:32.39
ETHER: Packet size = 74 bytes
ETHER: Destination = 0:0:c:2:78:36
ETHER: Source      = 0:0:c0:b8:c2:8d
ETHER: Ethertype = 0800
ETHER:
IP: ----- IP Header -----
IP:
IP: Version = 4
IP: Header length = 20 bytes
IP: Type of service = 0x00
IP:      xxx. .... = 0 (precedence)
IP:      ...0 .... = normal delay
IP:      .... 0... = normal throughput
IP:      .... .0.. = normal reliability
IP: Total length = 60 bytes
IP: Identification = 2947
IP: Flags = 0x0
IP:      .0.. .... = may fragment
IP:      ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live = 64 seconds/hops
IP: Protocol = 17
IP: Header checksum = c2ba
```

```
IP: Source address = 128.178.156.17
IP: Destination address = 128.178.25.8, IP: No options
IP:
UDP: ----- UDP Header -----
UDP:
UDP: Source port = 1267
UDP: Destination port = 53 (DNS)
UDP: Length = 40
UDP: Checksum = B672
UDP:
DNS: ----- DNS: -----
DNS:
DNS: ""
DNS:
```

```
ETHER: ----- Ether Header -----
ETHER:
ETHER: Packet 3 arrived at 19:03:32.40
ETHER: Packet size = 202 bytes
ETHER: Destination = 0:0:c0:b8:c2:8d, Western Digital
ETHER: Source = 0:0:c:2:78:36, Cisco
ETHER: Ethertype = 0800
ETHER:
IP: ----- IP Header -----
IP:
IP: Version = 4
IP: Header length = 20 bytes
IP: Type of service = 0x00
IP: xxx. .... = 0 (precedence)
IP: ...0 .... = normal delay
IP: .... 0... = normal throughput
IP: .... .0.. = normal reliability
IP: Total length = 188 bytes
IP: Identification = 38579
IP: Flags = 0x0
IP: .0.. .... = may fragment
IP: ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live = 58 seconds/hops
IP: Protocol = 17
IP: Header checksum = 3d0a
IP: Source address = 128.178.25.8,
IP: Destination address = 128.178.156.17,
IP: No options
IP:
UDP: ----- UDP Header -----
UDP:
UDP: Source port = 53
UDP: Destination port = 1267
UDP: Length = 168
UDP: Checksum = 0000
UDP:
DNS: ----- DNS: -----
DNS:
DNS: ""
DNS:
```



Q33/ What is the purpose for this packet exchange?

[A33]

Q34/ The length `Length = 168` refers to which portion of the packet?

[A34]

Q35/ Why do we see a difference of six between TTLs of the first packet and second packet? Write down any assumption you make?

[A35]

Q36/ What is the UDP port 1267 used for in this connection? Can you explain how is it allocated?

[A36]

5.2 IPV6 PACKET HEADERS

Padme is sitting in front of `lrcpc3` workstation and visits the website `'www.ethz.ch'`. An ethical hacker has read all the frames passing on the network. Here are the first two packets resulting from this activity:

```
ETHER: ----- Ether Header -----
ETHER:
ETHER: Packet 1 arrived at 11:55:22.298
ETHER: Packet size = 86 bytes
ETHER: Destination = 33:33:ff:01:00:01
ETHER: Source = 3c:07:54:3e:ab:f2
ETHER: Ethertype = 0x86dd
ETHER:
IP: ----- IP Header -----
IP:
IP: Version = 6
IP: Traffic class =0x00000000
IP:     .... 0000 00.. .... .... .... .... = Default Differentiated Service Field
IP:     ....     .... ..0. .... .... .... .... = No ECN-Capable Transport (ECT)
IP:     ....     .... ...0 .... .... .... .... = No ECN-CE
IP: .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
IP: Payload length = 32
IP: NextHeader= 58
IP: Hop limit= 255
IP: Source address = 2001:620:618:197:1:80b2:97c0:1
IP: Destination address = ff02::1:ff01:1
IP:
ICMPv6: ----- ICMPv6 Header -----
ICMPv6:
ICMPv6: Type = 135
ICMPv6: Code=0
ICMPv6: Checksum = 0xb199 [correct]
ICMPv6: Reserved = 00000000
ICMPv6: Target Address=2001:620:618:197:1:80b2:9701:1
ICMPv6:
```

```

ETHER: ----- Ether Header -----
ETHER:
ETHER: Packet 2 arrived at 11:55:22.306
ETHER: Packet size = 86 bytes
ETHER: Destination = 3c:07:54:3e:ab:f2
ETHER: Source = 00:08:e3:ff:fc:50
ETHER: Ethertype = 0x86dd
ETHER:
IP: ----- IP Header -----
IP:
IP: Version = 6
IP: Traffic class = 0x000000e0
IP:      .... 1110 00.. .... .... .... .... = Class Sector 7 Differentiated Service Field
IP:      ....      ..0. .... .... .... .... = No ECN-Capable Transport (ECT)
IP:      ....      ...0 .... .... .... .... = No ECN-CE
IP: .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
IP: Payload length = 32
IP: NextHeader=58 (ICMPv6)
IP: Hop limit= 255
IP: Source address = 2001:620:618:197:1:80b2:9701:1
IP: Destination address = 2001:620:618:197:1:80b2:97c0:1
IP:
ICMPv6: ----- ICMPv6 Header -----
ICMPv6:
ICMPv6: Type = 136
ICMPv6: Code=0
ICMPv6: Checksum = 0xe3f8 [correct]
ICMPv6: Flags=0xe0000000
ICMPv6:      1... .... .... .... .... = Router: Set
ICMPv6:      .1.. .... .... .... .... = Solicited: Set
ICMPv6:      ..1. .... .... .... .... = Override: Set
ICMPv6:      ...0 0000 0000 0000 0000 0000 0000 0000 = Reserved: 0
ICMPv6: Target Address=2001:620:618:197:1:80b2:9701:1
ICMPv6:

```



Q37/ What are these packets used for in the exchange? What do Types 135 and 136 in the ICMPv6 header represent?

[A37]

Q38/ Explain why the destination IP address in the first packet is `ff02::1:ff01:1`?

[A38]

Q39/ Which devices receive the first packet?

[A39]

Q40/ From the MAC layer, how can we confirm that both packets are indeed IPv6 packets?

[A40]

Q41/ Knowing that at EPFL IPv6 addresses are derived from IPv4 addresses, can you guess the IPv4 addresses of the source and destination of the second packet ?

[A41]