TÉCNICO LISBOA

RC 17/18	LAB ASSIGNMENT	Number:	7
Wireshark Lab: Protocol Analysis		Issue Date:	6 Nov 2017
Internet Protocol (IP)		Due Date:	17 Nov 2017

Preliminary Notes

The instructions in this document are applicable to Computers at the IST Labs, which are already configured with the required tools. Nevertheless, one nice feature of the software stack we are going to use is that it is portable to many platforms including **YOUR OWN** personal computers. The instructions are suitable for machines running the following Operating Systems:

- Microsoft Windows from version 7 up to 10
- Apple macOS from versions 10.8 'Mountain Lion' up to 11.13 'High Sierra'
- Debian-based Linux, such as Ubuntu (recommended) from versions 12.04 'Precise' to 16.04 'Xenial Xerus'.

Note: Avoid copying text strings from the command line examples or configurations in this document, as pasting them in your system or files may introduce/modify some characters, leading to errors or inadequate results.

1 Intro

In this lab, we'll take a quick look at the IP. Internet Protocol is responsible for relaying datagrams across network boundaries. We will have a look at IPv4 and IPv6.

2 Experimental Environment

Run up your C.O.R.E. machine and access to the GUI by issuing:

```
:~$ vagrant ssh -c core-gui -- -X
```

In C.O.R.E. we need to create a simple network between two machines and start your simulation. This simple network will consist of one PC and one Server connected by a router, as illustrated in Figure 1. In order to create you network use the tools available at the left toolbar.

RC 17/18	LAB ASSIGNMENT	Number:	7
Wireshark Lab: Protocol Analysis		Issue Date:	6 Nov 2017
Internet Protocol (IP)		Due Date:	17 Nov 2017

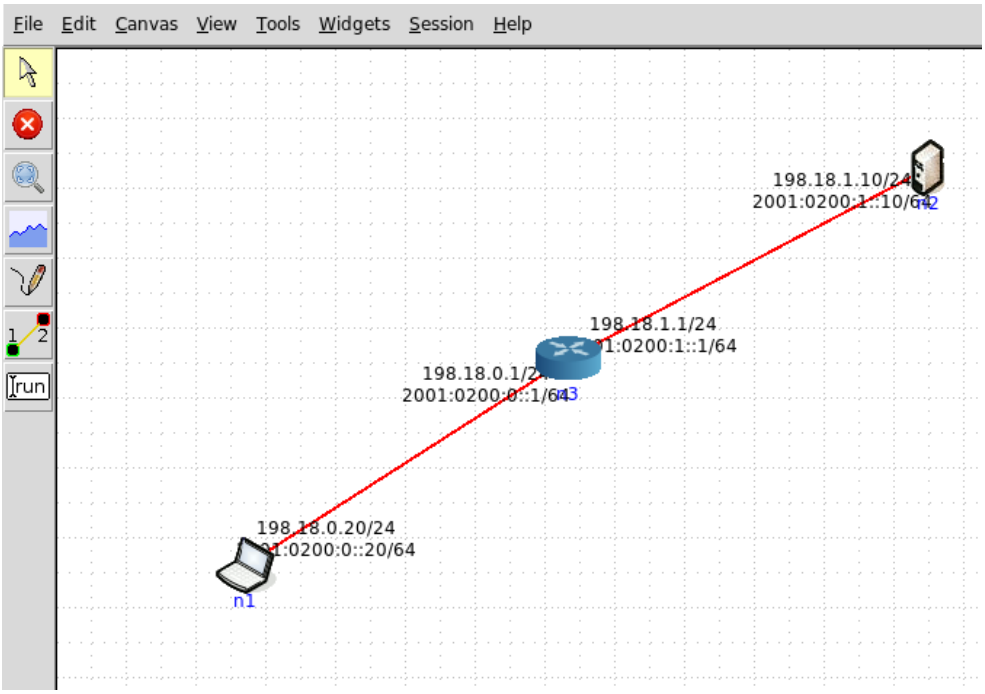

Figure 1: Simple network created on C.O.R.E.

In order to analyze the packets exchanged between, we will use the Wireshark. Open one Wireshark window in one of the end systems:

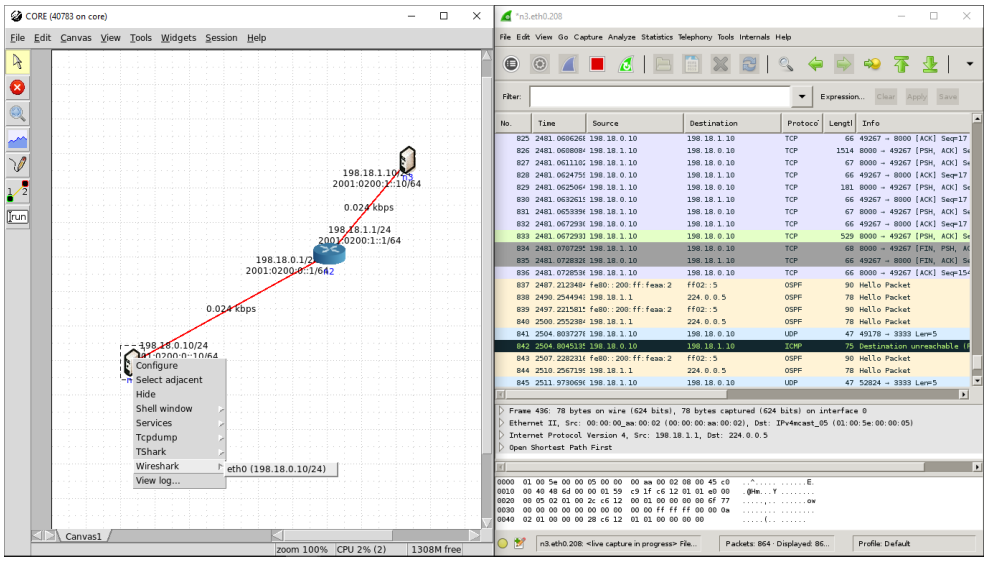

Figure 2: Wireshark window example

In order to have a realistic you should double click in the connection and add some delay and lost probability.

2.1 IPv4 Analysis

In order to analyze one IPv4 packet we will use ping to transfer messages. Open one bash window in one of the end systems.

Issue the following command

```
root@n1:~$ ping <ipv4-address>
```

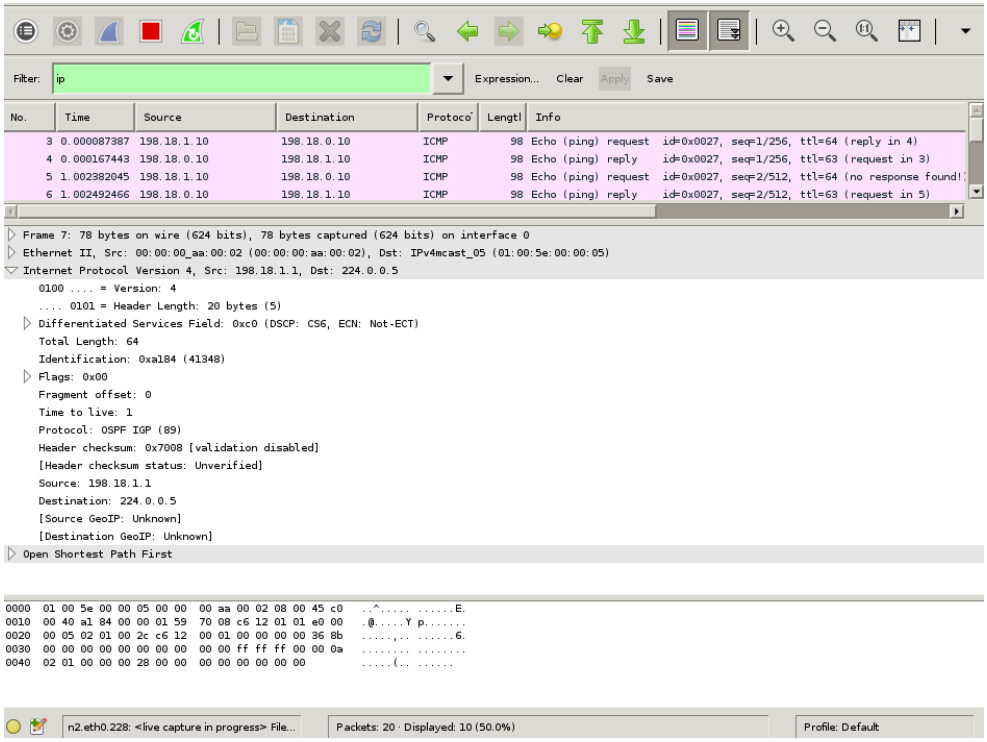

Figure 3: Wireshark IPv4 example

We should get some IP packets in our wireshark capture. Select the first ICMP message sent by your computer, and expand the Internet Protocol part of the packet in the packet details window. You can use the filter in order to isolate IPv4 packets, by filling ”ip”:

To print a packet, use File $\rightarrow$ Print, choose Selected packet only, choose Packet summary line, and select the amount of packet detail.

Questions:

1. What is the IPv4 address of your computer?
2. Within the IPv4 packet header, what is the value in the upper layer protocol field?
3. How many bytes are in the IPv4 header? How many bytes are in the payload of the IPv4 datagram? Explain how you determined the number of payload bytes.
4. Has this IPv4 datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.

2.2 IPv6 Analysis

In order to analyze one IPv6 packet we will use ping6 to transfer messages. We will send messages with a size of 5000 bytes. Open one bash window in one of the end systems.

Issue the following command

```
root@n1:~$ ping6 -s 5000 <ipv6-address>
```

In our Wireshark we will have a lot of IPv6 packets. You can use the filter in order to isolate IPv6 packets, by filling ”ipv6”:

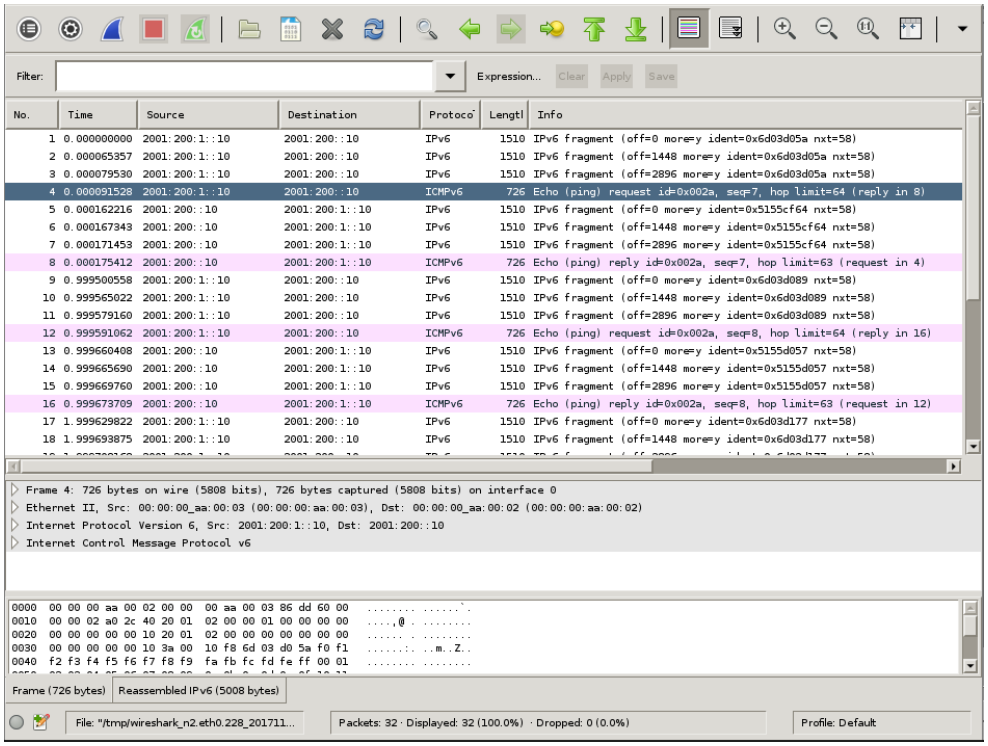

Figure 4: Wireshark IPv6 example

Questions

1. What is the IPv6 address of your computer?
2. Within the IPv6 packet header, what is the value in the upper layer protocol field?
3. How many bytes are in the IPv6 header? How many bytes are in the payload of the IPv6 datagram? Explain how you determined the number of payload bytes.
4. Has this IPv6 datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.
5. How many fragments were created from the original datagram?
6. What fields change in the IP header among the fragments?

3 Finishing your Experiments

In order to stop the Virtual Machines and to verify the global state of all active Vagrant environments on the system, we can issue the following commands:

Confirm that the statuses of the VMs is 'powered off'.

In Lab computers, also do the following (answering "y"), in order to ensure the system is cleaned from your experime nts. Note however that the configuration files remain stored in your private area in AFS:

```
:~$ vagrant destroy
default: Are you sure you want to destroy the 'default' VM? [y/N]
==> default: Destroying VM and associated drives...
:~$ vagrant global-status
```

Confirm that there are no VMs listed.

4 Report the results

The experiences in LAB#6 and in this LAB#7 are for evaluation.

4.1 Lab Report

Create a Google Doc Lab Report using one of the Google Docs "Science Report" Template. In the report, explain what you did in both in **LAB#6** (UDP and TCP) and in this **LAB#7** (IP), and include the relevant screen shots with captions, the text results of commands or wireshark captures of the experiments, explaining them.

Submission of your work:

Filename:
Report MUST be named **RC1718-LAB07-Report-IDofStudent-IDofGroup** where "**IDofStudent**" must have the format "ist123456" and "**IDofGroup**" must have the format "Gxx" where "xx" is the number of the Group.

Submission Form:
Your work is submitted in a Google Form available from the Assignment published in Google Classroom.

Due Date:
Please respect the due date of the assignment, considering the time limit of 23:59h.

WARNING. Incorrect submissions will not be considered for evaluation.