TI305: Network architecture

TP2: Capture and Analysis of Network Frames with Wireshark

Duration: 2 hours

Instructions:

⇒ This work is INDIVIDUAL. It "**looks long**", but is very rewarding when you read the instructions correctly and follow the instructor's explanations.

⇒ The first part is an introduction and seems very theoretical, but mastery of Wireshark and frame analysis comes with experience.

⇒ There are no unnecessary questions, as long as they help you progress.

System requirements: PC, Wireshark.

Learning outcomes :

At the end of this session, each student should be able to :

- Understand the basics of frame analysis with Wireshark.
- Apply filters to display and analyze network traffic.
- Identify the key elements of network frames.
- Differentiate network communications across multiple subnets.

Part 1: Guided tour

Step 1: Getting started with Wireshark

1.1. Introduction to Wireshark

Wireshark is an open-source network protocol analysis software created by Gerald Combs in 1998. An international group of network experts and developers now manage the tool and update it to ensure compatibility with new network technologies and encryption methods. Wireshark poses absolutely no security risk.

1.2. Installing Wireshark

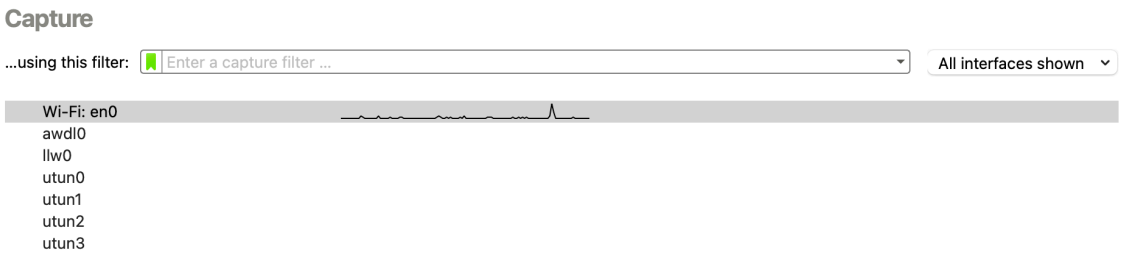
Downloading and installing Wireshark couldn't be simpler. First step: visit the official Wireshark download page (https://www.wireshark.org/download.html) and find the version corresponding to your operating system. The standard edition of the tool is free.

If you haven't already done so, install Wireshark on your machine.

1.3. Launch of Wireshark

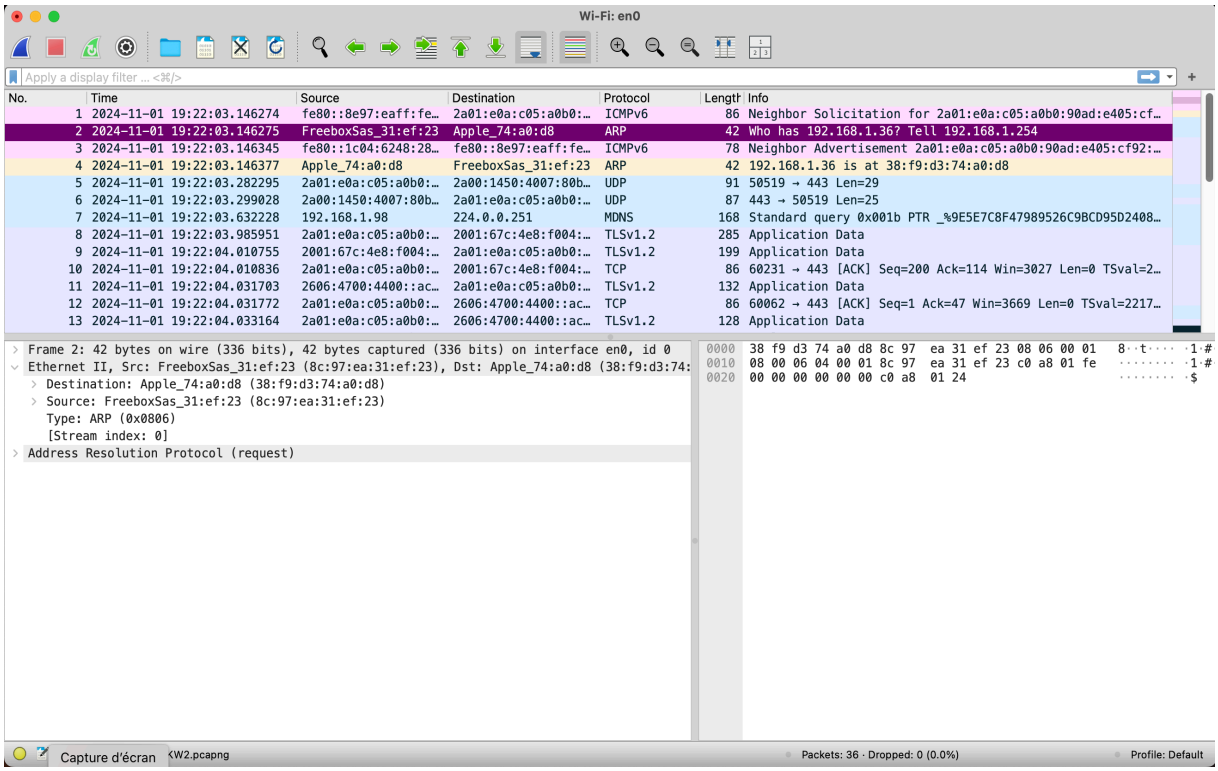
To avoid capturing on the school network (too much traffic and not always audible to a beginner), your teacher will create a wireless network, without Internet access, on which this discovery part will take place.

Once connected to the TP wireless network, launch the Wireshark application and select the Wi-Fi interface as shown in the following figure, then start capturing.



1.4. Interface overview

Now let yourself be carried away by your operator's explanations of the interface.



Step 2: Apply Filters

You've captured thousands of frames in just a few minutes. Wireshark captures everything that passes through your network.

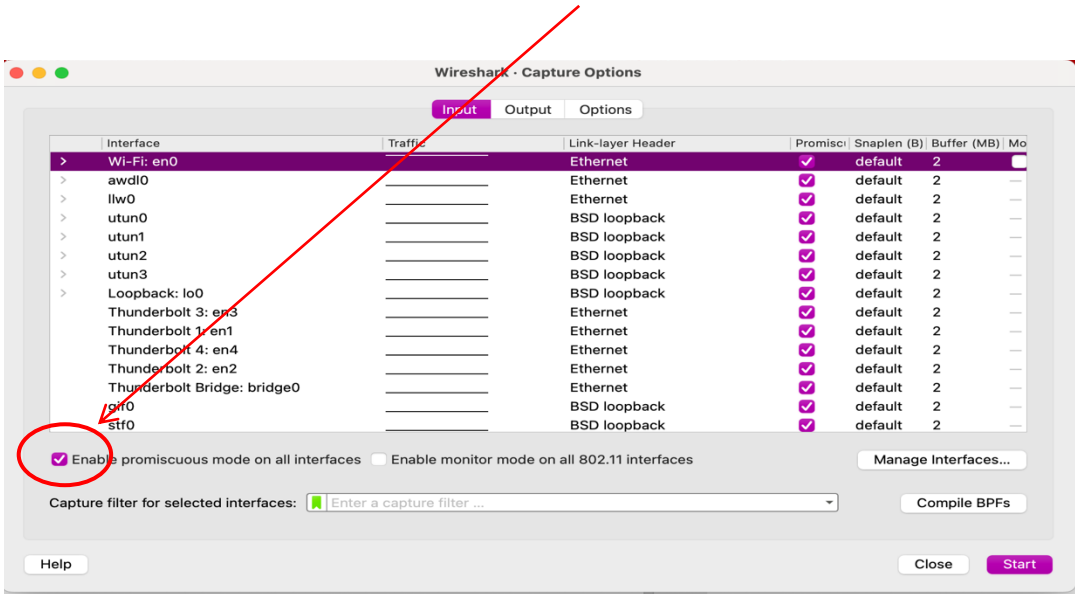
Capture filters and display filters are among Wireshark's most useful features. They allow you to choose the most effective display mode for solving your problems. Here are a few examples of filters to get you started.

- **Ethernet filters** :
    - Display Ethernet frames only: eth
    - Display frames with a specific MAC address: eth.addr == xx:xx:xx:xx:xx:xx:xx
- **Filters by IP address** :
    - Show only traffic from a specific IP address: ip.src == 192.168.1.1
    - Display traffic directed to a specific IP address: ip.dst == 192.168.1.2
- **HTTP filters** :
    - Display only HTTP packets: http
    - Display HTTP GET traffic: http.request.method == "GET"
    - Display HTTP responses: http.response
- **ICMP filters** :
    - Display ICMP packets only: icmp
    - Display ping requests: icmp.type == 8
    - Display ping responses: icmp.type == 0
- **DNS filters** :
    - Display DNS requests only: dns.qry.name
    - Display DNS responses only: dns.response
- **TCP filters** :
    - Display TCP packets: tcp
    - Display established TCP connections: tcp.flags.syn == 1 && tcp.flags.ack == 1
- **UDP filters** :
    - Display UDP packets only: udp

Notes :

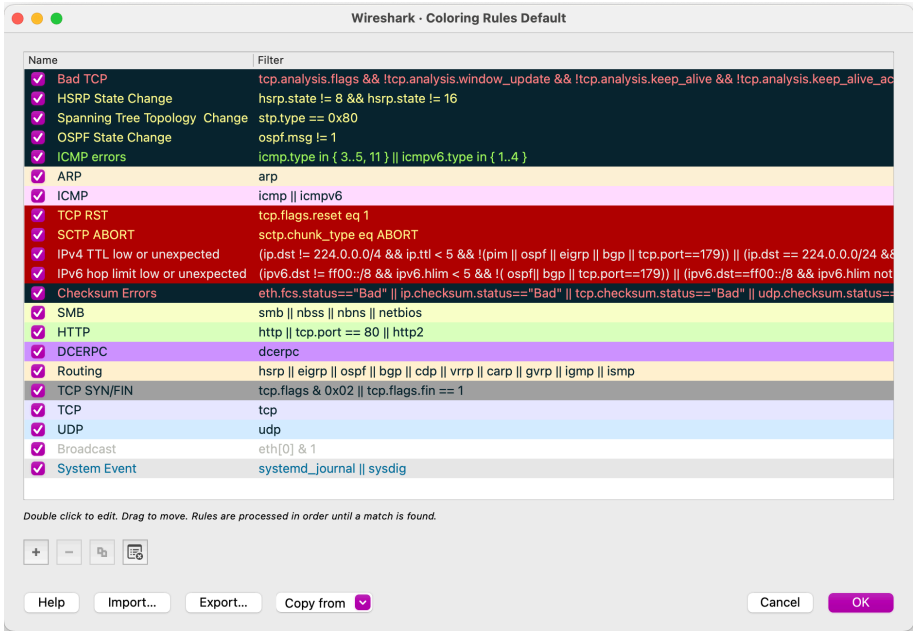
Wireshark's Promiscuous mode

By default, Wireshark only captures packets sent and received by the computer on which it is running. By checking the ***Promiscuous Mode*** *box in the capture settings, you can capture most of the traffic occurring on the local network.*



Wireshark coloring rules

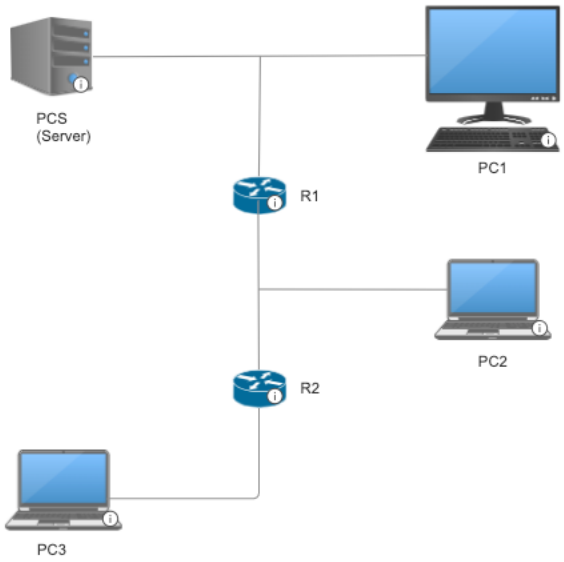
First of all, you'll notice that packets are highlighted in a variety of colors. Wireshark uses colors to help identify traffic types. These colors help to visually scan the list of packets to identify problems, traffic types and protocols used. You can also customize Wireshark's coloring rules to suit your analysis needs. To do this, go to ***View > Coloring Rules*** *and modify or create new rules as necessary.*



PART 2: Unguided

For this step, download the capture folder found in the same section as this TP. Once downloaded, you can unzip it. It contains captures that we'll be using for the rest of this tutorial.

Let's say the following network (the rest of the network information is to be discovered via captures):



***Notes** :*

Boxes R1 and R2 represent routers linking different local networks: machines PC1 and PCS are not on the same local network as machine PC2, which is not on the same local network as machine PC3 either.

The PCS machine contains a super-intelligent server that responds to all queries. The aim of this tutorial is to look at the frames that circulate on the network when PC1, PC2 and PC3 interrogate the server.

Stage 1: Discovery

At this stage, the aim is to look at and understand the capt1.pcapng frames.

You therefore need to launch Wireshark and open this capture to discover its contents.

This screenshot shows the launch of the client on machine PC1, which queries the server to answer all its questions.

He has 4 packets in the capture: we're going to look at packets 3 and 4. The first two packets will be useful later.

- Try to guess the IP address of PCS and PC1 from this packet (Help: in client-server communication, it's usually the client who talks to the server first).

- Do the same with package number 4.
- Find out what the server said to the client.

Step 2: Broadcasting messages

At this stage, the aim is to look at and understand the frames of capture capt2.pcapng. This capture was obtained when two clients, both present on machine PC1, sent a question to the server.

- Look at the 4 frames sent, and in particular the first two. Find out how the server can differentiate between the frames and see that they are from two different clients. Find out how to match the questions with the answers.

Step 3: Further analysis

Let's take a look at what happens when a client on machine PC2 talks to the server. We have access to two different captures: capt3.pcapng and capt4.pcapng. One was taken on the top network, the other on the middle network.

We only look at the last two frames of each capture (the one where UDP is written), and we don't bother for the moment with the ones where ARP is written.

- Find the similarities and differences between the frames in capt3.pcapng and capt4.pcapng
- Find PC2's IP address
- Ethernet addresses, also known as MAC addresses, are the addresses of machines' network interfaces. The Ethernet layer therefore explains, on the same local network, from which network interface the frame originated, and to which network interface it is destined. By comparing these two frames, as well as the frames from the very first capture, find the MAC addresses of PC1, PC2, PCS and the two MAC addresses (one per interface) of router R1.

Now look at the ARP frames observed in steps 1 and 3.

- What can they be used for? (Help: ff:ff:ff:ff:ff:ff is a special MAC address that means "send to the whole network", also known as broadcast MAC).
- Deduce the two IP addresses of router R1.

Step 4: I've found everything, so I've understood everything

This is the final step in the discovery process. Here, PC3 talks to the server. We have access to three different captures: capt51.pcapng, capt52.pcapng and capt53.pcapng. One of the captures was made in the network above, another in the middle network, and the third in the network below, but you don't know which network each of the captures corresponds to.

Look at the 3 captures:

- Find the MAC address and IP address of PC3, as well as the two MAC addresses of router R2.
- Find out which capture was made on the top network, which was made on the middle network, and which was made on the bottom network.
- Look closely at the network layer (IP) of the frames and find out what's changing.

If you've been able to answer all the questions, you've got a good basis for frame analysis.