



# NetPractice

*Summary: Discover the basics of networking.*

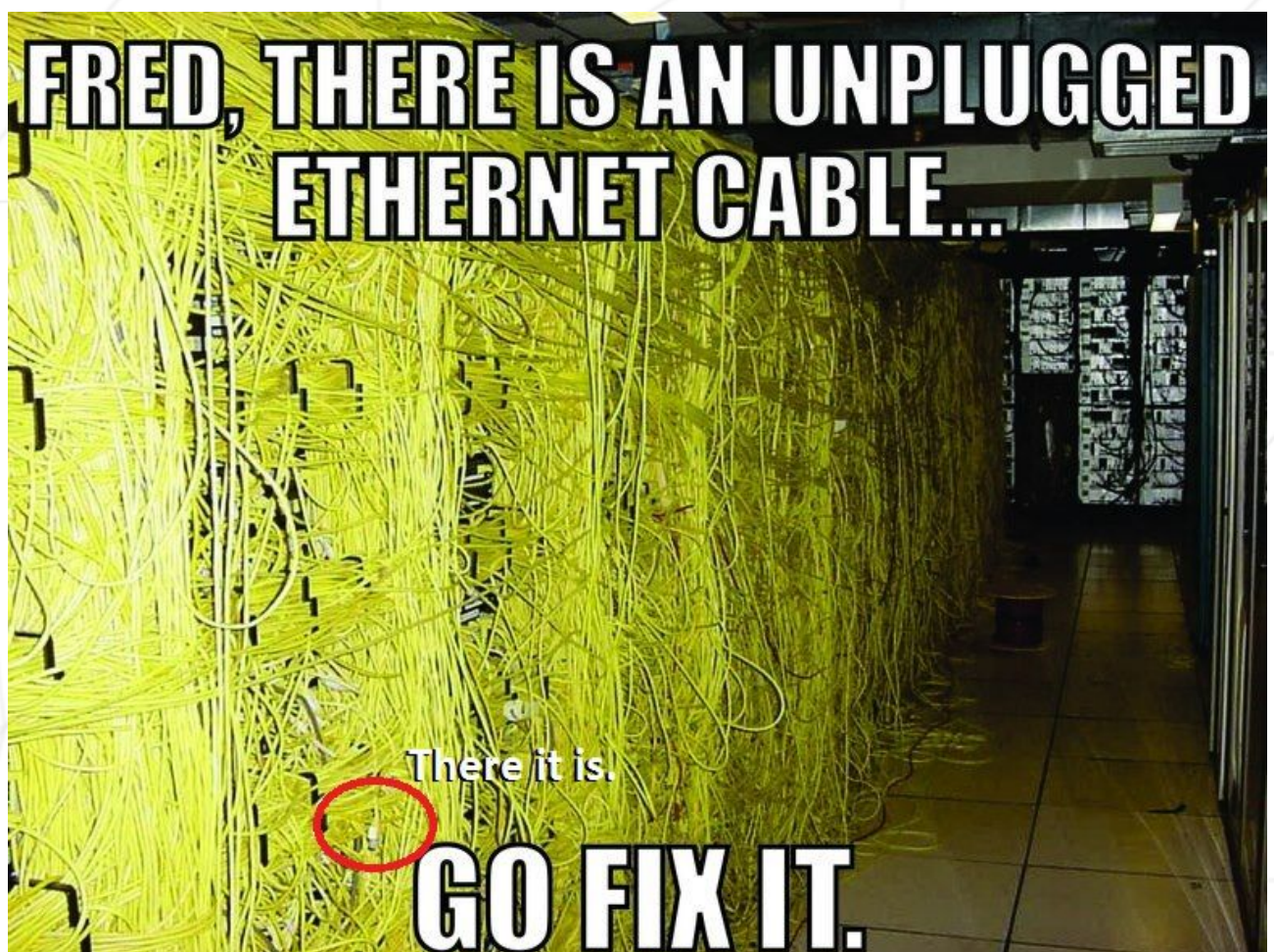
*Version: 4.0*

# Contents

<b>I</b>	<b>Preamble</b>	<b>2</b>
<b>II</b>	<b>Introduction</b>	<b>3</b>
<b>III</b>	<b>General guidelines</b>	<b>4</b>
<b>IV</b>	<b>AI Instructions</b>	<b>5</b>
<b>V</b>	<b>Mandatory part</b>	<b>7</b>
<b>VI</b>	<b>Submission and peer-evaluation</b>	<b>10</b>

# Chapter I

## Preamble



# **Chapter II**

## **Introduction**

This project is a general practical exercise designed to help you discover networking.

# Chapter III

## General guidelines

You will need to configure small-scale networks. To do so, it is necessary to understand how TCP/IP addressing functions.

You will need to complete 10 levels (i.e. 10 exercises) and submit them to your **Git** repository.



In this project, the networks you will work with are simulated. They will be available via a training interface that you will open in your web browser.

# Chapter IV

## AI Instructions

### ● Context

During your learning journey, AI can assist with many different tasks. Take the time to explore the various capabilities of AI tools and how they can support your work. However, always approach them with caution and critically assess the results. Whether it's code, documentation, ideas, or technical explanations, you can never be completely sure that your question was well-formed or that the generated content is accurate. Your peers are a valuable resource to help you avoid mistakes and blind spots.

### ● Main message

- 👉 Use AI to reduce repetitive or tedious tasks.
- 👉 Develop prompting skills — both coding and non-coding — that will benefit your future career.
- 👉 Learn how AI systems work to better anticipate and avoid common risks, biases, and ethical issues.
- 👉 Continue building both technical and power skills by working with your peers.
- 👉 Only use AI-generated content that you fully understand and can take responsibility for.

### ● Learner rules:

- You should take the time to explore AI tools and understand how they work, so you can use them ethically and reduce potential biases.
- You should reflect on your problem before prompting — this helps you write clearer, more detailed, and more relevant prompts using accurate vocabulary.
- You should develop the habit of systematically checking, reviewing, questioning, and testing anything generated by AI.
- You should always seek peer review — don't rely solely on your own validation.

## ● Phase outcomes:

- Develop both general-purpose and domain-specific prompting skills.
- Boost your productivity with effective use of AI tools.
- Continue strengthening computational thinking, problem-solving, adaptability, and collaboration.

## ● Comments and examples:

- You'll regularly encounter situations — exams, evaluations, and more — where you must demonstrate real understanding. Be prepared, keep building both your technical and interpersonal skills.
- Explaining your reasoning and debating with peers often reveals gaps in your understanding. Make peer learning a priority.
- AI tools often lack your specific context and tend to provide generic responses. Your peers, who share your environment, can offer more relevant and accurate insights.
- Where AI tends to generate the most likely answer, your peers can provide alternative perspectives and valuable nuance. Rely on them as a quality checkpoint.

### ✓ Good practice:

I ask AI: "How do I test a sorting function?" It gives me a few ideas. I try them out and review the results with a peer. We refine the approach together.

### ✗ Bad practice:

I ask AI to write a whole function, copy-paste it into my project. During peer-evaluation, I can't explain what it does or why. I lose credibility — and I fail my project.

### ✓ Good practice:

I use AI to help design a parser. Then I walk through the logic with a peer. We catch two bugs and rewrite it together — better, cleaner, and fully understood.

### ✗ Bad practice:

I let Copilot generate my code for a key part of my project. It compiles, but I can't explain how it handles pipes. During the evaluation, I fail to justify and I fail my project.



# Chapter V

## Mandatory part

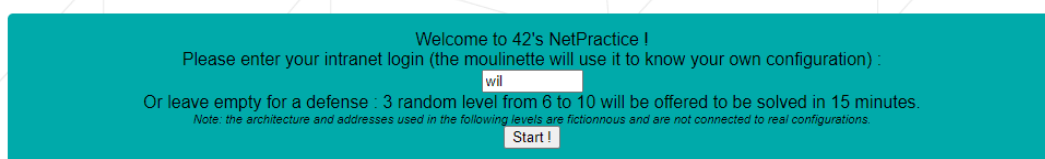
This project involves solving networking problems to ensure a network operates correctly.

First, download the file attached to the project page.

Then, extract the files into any folder you choose.

In this folder, open the `index.html` file.

This interface should launch in your web browser:

A screenshot of a web interface for '42's NetPractice'. It features a teal background with white text. At the top, it says 'Welcome to 42's NetPractice !'. Below that, it asks the user to enter their intranet login, with a small text box containing 'will' and a password field. A note mentions that leaving the field empty will result in a defense with 3 random levels from 6 to 10, to be solved in 15 minutes. A 'Start !' button is at the bottom.

Welcome to 42's NetPractice !  
Please enter your intranet login (the moulinette will use it to know your own configuration) :  
Or leave empty for a defense : 3 random level from 6 to 10 will be offered to be solved in 15 minutes.  
Note: the architecture and addresses used in the following levels are fictionnous and are not connected to real configurations.  
Start !

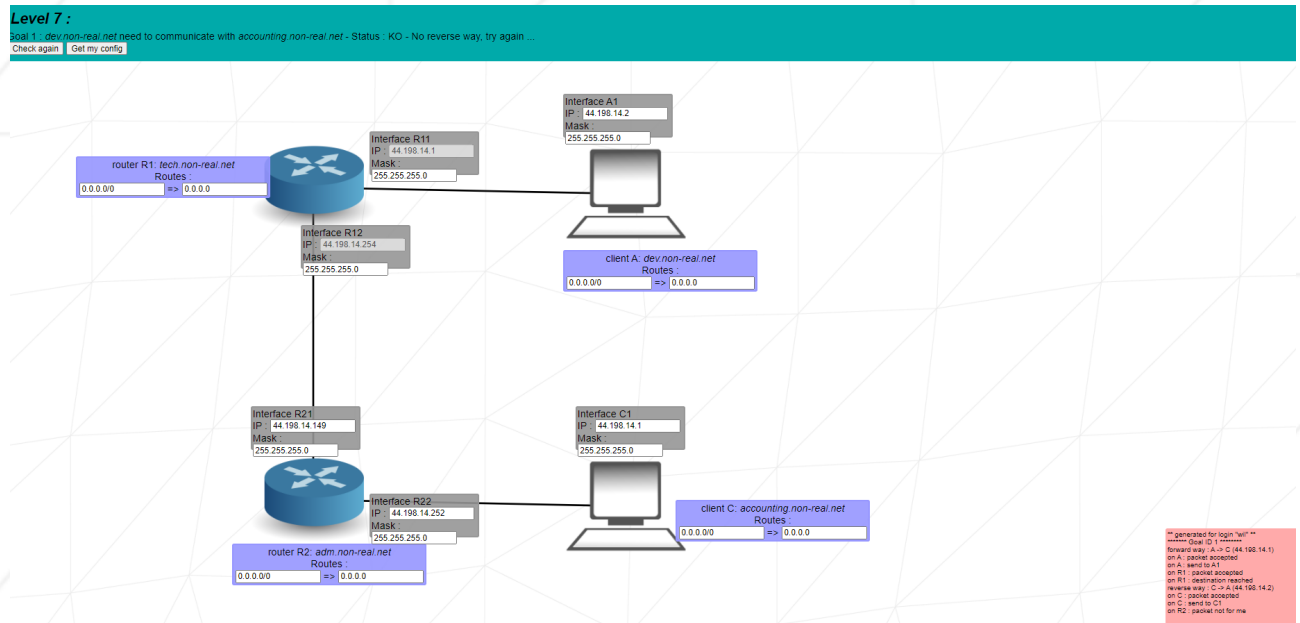
Welcome to NetPractice! :)

As mentioned on the page:

- You can practice by entering your login in the field
- Alternatively, you can try the 'correction' version by leaving the field empty.



There are 10 levels available for training. Below is an example:



For each level, a non-functional network diagram will appear.

At the top of your window, you will see a goal to achieve, which outlines the issues to solve for the network to function properly. There are two buttons you can use:

- **[Check again]** to verify if your configuration is correct or not.
- **[Get my config]** to download your configuration whenever necessary. It will be useful to turn in your assignment.

### Level 7 :

Goal 1 : dev.non-real.net need to communicate with accounting.non-real.net - Status : KO - No reverse way, try again ...

[Check again](#) [Get my config](#)

Upon successfully completing a level, a new button will appear. Click on this button to get to the next level.

[Check again](#) [Get my config](#) [Next level](#)



Before proceeding to the next level, ensure you export your configuration using the **[Get my config]** button so you can put it in your Git repository.

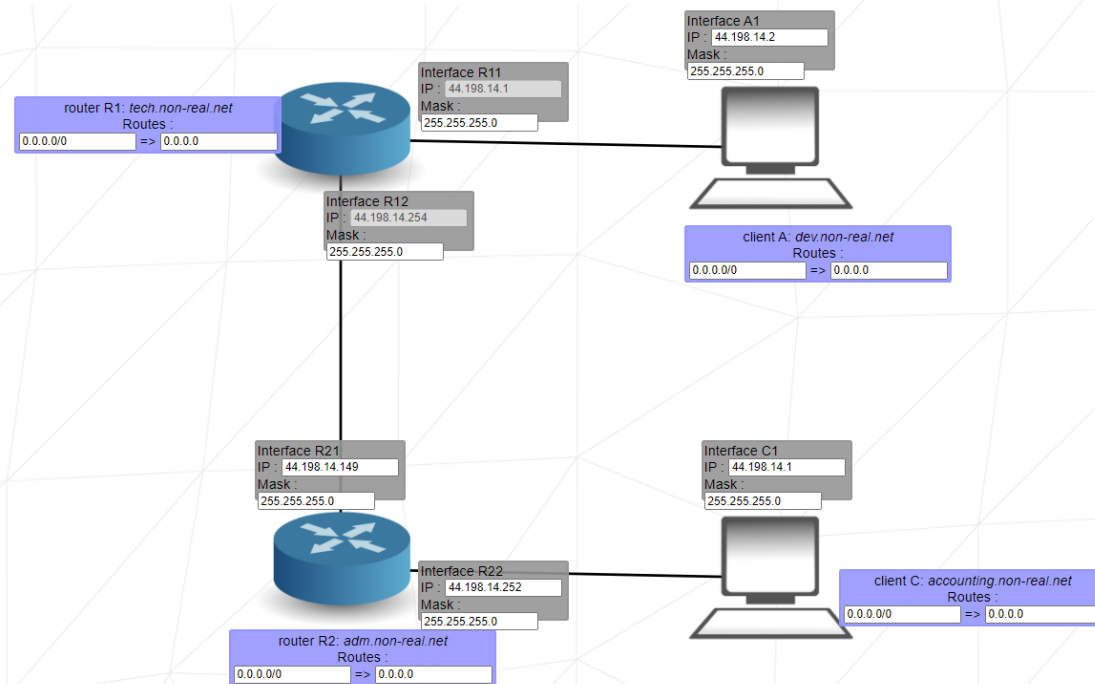
At the bottom of the page, logs are displayed. They can be helpful to understand why your configuration is not functioning.

```

** generated for login "wil" **
***** Goal ID 1 *****
forward way : A -> C (44.198.14.1)
on A : packet accepted
on A : send to A1
on R1 : packet accepted
on R1 : destination reached
reverse way : C -> A (44.198.14.2)
on C : packet accepted
on C : send to C1
on R2 : packet not for me

```

Here is an example of what kind of exercise you will get:



To succeed, adjust the unshaded fields until your network configuration is correct and functional.

To complete this assignment, it is strongly recommended that you understand how addressing functions in a network containing devices such as routers. Familiarize yourself with TCP/IP addressing.

# Chapter VI

## Submission and peer-evaluation

Submit your assignment in your `Git` repository as usual. Only the work within your repository will be evaluated during the defense. Do not hesitate to double-check the names of your files to ensure they are correct (if relevant).

Since 10 levels are available in the training interface, you will need to submit 10 files to your repository (1 file per level). Place them at the root of your repository.

Do not forget to enter your login in the training interface. Export a file for each level using the `[Get my config]` button.



It is very important that you enter your login in the interface.

During the defense, you will need to successfully complete 3 random levels as mentioned on the training platform. You will have a limited time to do so: 15 minutes.



You are prohibited from using external tools during your evaluation.  
The use of a simple calculator, such as `"bc"`, is tolerated.

During the evaluation, a brief **modification of the project** may occasionally be requested. This could involve a minor behavior change, a few lines of code to write or rewrite, or an easy-to-add feature.

While this step may **not be applicable to every project**, you must be prepared for it if it is mentioned in the evaluation guidelines.

This step is meant to verify your actual understanding of a specific part of the project. The modification can be performed in any development environment you choose (e.g., your usual setup), and it should be feasible within a few minutes — unless a specific

timeframe is defined as part of the evaluation.

You can, for example, be asked to make a small update to a function or script, modify a display, or adjust a data structure to store new information, etc.

The details (scope, target, etc.) will be specified in the **evaluation guidelines** and may vary from one evaluation to another for the same project.



????????????? XXXXXXXXXX = \$\$\$84f5ef7ed1616b2fbc6c35e9567d0075