

# Le modèle en couches et l'identification d'un protocole

Tout ce dont vous avez besoin pour ce module, dans un seul document. Cochez au fur et à mesure.

**U4-S1-M2**

module

**Élève**

document

**MaisonDesJeux**

scénario 1

**B · 3 h**

format et durée

|                                                             |           |
|-------------------------------------------------------------|-----------|
| <b>Ce qu'il faut savoir avant de commencer</b>              | <b>4</b>  |
| C'est quoi un client et un serveur ?                        | 4         |
| C'est quoi une adresse IP ?                                 | 4         |
| C'est quoi une adresse MAC ?                                | 4         |
| Comment lire des octets ?                                   | 4         |
| Vérifiez-vous                                               | 5         |
| <b>Le modèle en couches</b>                                 | <b>6</b>  |
| Capsule 1 sur 4 Pourquoi des couches ?                      | 6         |
| Capsule 2 sur 4 Le modèle TCP/IP et la place des protocoles | 9         |
| Capsule 3 sur 4 L'encapsulation                             | 11        |
| Capsule 4 sur 4 Reconnaître un protocole dans une capture   | 13        |
| Ce qu'on retient                                            | 16        |
| <b>Le modèle en couches</b>                                 | <b>17</b> |
| L'essentiel                                                 | 17        |
| Les mots à connaître                                        | 17        |
| Les deux modèles                                            | 18        |
| Les ports à connaître                                       | 18        |
| La méthode                                                  | 18        |
| Un exemple concret                                          | 18        |
| Les pièges                                                  | 19        |
| <b>Lire une capture couche par couche</b>                   | <b>20</b> |
| PARTIE A — Voir les couches d'une trame                     | 21        |
| PARTIE B — Annoter une trame couche par couche              | 25        |
| PARTIE C — Nommer le protocole des six captures             | 29        |
| PARTIE D — En clair ou chiffré, et qui demande ?            | 31        |
| POUR ALLER PLUS LOIN                                        | 36        |
| CE QU'IL FAUT RETENIR                                       | 37        |
| <b>Tous les documents du module</b>                         | <b>38</b> |
| A1 — Les six captures à ouvrir dans Wireshark               | 38        |
| A2 — Extrait du guide de Wireshark                          | 39        |

|                                                |    |
|------------------------------------------------|----|
| A3 — La trame à annoter                        | 41 |
| A4 — Les six captures imprimées                | 42 |
| A5 — La table des ports usuels                 | 51 |
| A6 — Les machines de la maquette MaisonDesJeux | 52 |

# Ce qu'il faut savoir avant de commencer

*Ce qu'il faut savoir avant de commencer. Lisez-le si vous avez un doute. Environ 10 minutes.*

## C'est quoi un client et un serveur ?

Un **client** est le programme qui demande. Un **serveur** est le programme qui répond.

Exemple : votre navigateur demande une page. Le serveur web la renvoie. Le client parle toujours en premier.

## C'est quoi une adresse IP ?

Une adresse IP est le numéro d'une machine dans le réseau. Elle s'écrit avec quatre nombres de 0 à 255, séparés par des points.

Exemple : 192.168.10.20. Deux machines du même réseau n'ont jamais la même adresse IP.

## C'est quoi une adresse MAC ?

Une adresse MAC est le numéro de la carte réseau d'une machine. Elle s'écrit avec six paires de caractères, séparées par deux-points.

Exemple : 02:4d:44:4a:00:20. Elle sert à livrer les données à la bonne machine, dans le réseau local.

## Comment lire des octets ?

Un **octet** est un petit paquet de 8 bits : 8 chiffres 0 ou 1. On l'écrit avec deux caractères, de 00 à ff.

Les caractères vont de 0 à 9, puis de a à f. C'est l'**hexadécimal**.

Un relevé d'octets montre 16 octets par ligne. Vous ferez seulement quelques additions et soustractions : vous allez surtout compter et encadrer des octets.

## Vérifiez-vous

---

☐

1. Vous ouvrez une appli de météo sur votre téléphone. Qui est le client ? Qui est le serveur ?

---

☐

2. Voici deux écritures : `192.168.10.101` et `02:4d:44:4a:01:01` . Quelle est l'adresse IP ?

---

☐

3. Combien de caractères faut-il pour écrire un octet en hexadécimal ?

---

## Réponses

1. Le client est l'appli du téléphone : elle demande la météo. Le serveur est la machine de l'entreprise de météo : il répond.
2. `192.168.10.101` . L'autre est une adresse MAC.
3. Deux caractères, de `00` à `ff` .

# Le modèle en couches

COURS-U4-S1-02

Marc Oliveira est le trésorier de MaisonDesJeux, une association de jeux de société. Il garde le fichier des adhérents sur le serveur de l'association.

Bruno, un bénévole, a installé un logiciel qui enregistre tout ce qui passe sur le réseau. Il montre l'écran à Marc : des centaines de lignes, des chiffres, des noms bizarres.

« Regarde, on voit tout passer. Ton fichier aussi. »

Marc est inquiet. Il vous demande : « Est-ce que n'importe qui peut lire mon fichier sur le réseau ? »

À la fin de la séance, vous saurez lire un enregistrement du réseau couche par couche. Vous saurez dire quel service passe, qui le demande, et si on peut le lire.

## LE PLAN DE LA SÉANCE

- 1 Pourquoi des couches ?
- 2 Le modèle TCP/IP et la place des protocoles.
- 3 L'encapsulation.
- 4 Reconnaître un protocole dans une capture.

## LA QUESTION DE DÉPART

Vous envoyez un message vocal à un ami. Votre téléphone passe par le Wi-Fi de la maison, puis par Internet, puis par le réseau 4G de votre ami. Est-ce que votre appli de messagerie doit savoir tout ça ?

## CAPSULE 1 SUR 4

### Pourquoi des couches ?

Envoyer des données, c'est beaucoup de travaux différents : choisir le service, trouver la machine, trouver le chemin, transformer les bits en signal. Comment les organiser ?

Deux machines qui se parlent suivent des règles communes. Ces règles s'appellent un **PROTOCOLE** : une langue que les deux côtés comprennent. Chaque protocole fait un seul travail.

On range les protocoles en étages, les **COUCHES** : chaque couche rend service à la couche du dessus, et utilise celle du dessous.

L'intérêt est de pouvoir changer une pièce sans tout casser. Vous passez du câble au Wi-Fi : seule la couche du bas change. Votre navigateur ne voit aucune différence.

### INSTANT ANALOGIE

Pensez à un colis envoyé par la poste. Vous écrivez une lettre et vous la mettez dans une enveloppe.

La poste ne lit pas la lettre : elle lit seulement l'adresse. Le camion ne lit même pas l'adresse : il transporte des sacs.

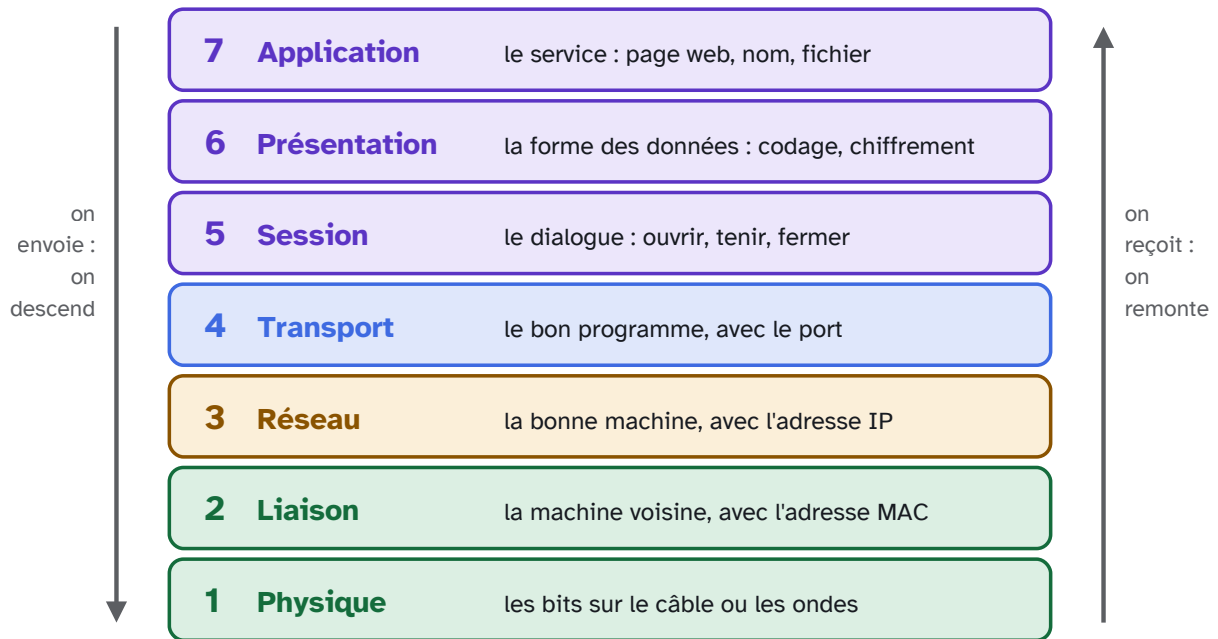
Chacun fait son travail, sans s'occuper du travail des autres.

L'image s'arrête là. Sur un réseau, chaque couche ajoute sa propre enveloppe, et il y en a quatre les unes dans les autres. Et ces enveloppes ne sont pas fermées : sans chiffrement, n'importe qui sur le chemin peut lire la lettre.

Le modèle de référence s'appelle le **MODÈLE OSI**, pour *Open Systems Interconnection* : interconnexion de systèmes ouverts. Il découpe le travail en sept couches, numérotées de 1 en bas à 7 en haut.

À l'envoi, les données descendent les couches. À l'arrivée, elles les remontent.

| N° | Couche              | Son travail                                                         |
|----|---------------------|---------------------------------------------------------------------|
| 7  | <b>Application</b>  | le service lui-même : page web, nom de site, fichier                |
| 6  | <b>Présentation</b> | la forme des données : codage, chiffrement                          |
| 5  | <b>Session</b>      | le dialogue : l'ouvrir, le tenir, le fermer                         |
| 4  | <b>Transport</b>    | livrer au bon programme, grâce au port, et vérifier que tout arrive |
| 3  | <b>Réseau</b>       | trouver la bonne machine, même loin, grâce à l'adresse IP           |
| 2  | <b>Liaison</b>      | livrer à la machine voisine, grâce à l'adresse MAC                  |
| 1  | <b>Physique</b>     | transformer les bits en signal, sur le câble ou dans les ondes      |



**Schéma** • Les sept couches du modèle OSI : on descend pour envoyer, on remonte pour recevoir

#### ON LE FAIT ENSEMBLE — VOUS REGARDEZ UNE VIDÉO EN WI-FI

1. Quelle couche transforme les bits en ondes radio ?
2. Quelle couche livre les données à la box, la machine voisine ?
3. Quelle couche trouve le serveur de vidéos, très loin ?
4. Quelle couche livre les images au bon programme, le navigateur, et pas au jeu ouvert à côté ?

#### Exemple

#### Pourquoi

|   |                                                                      |                                                              |
|---|----------------------------------------------------------------------|--------------------------------------------------------------|
| ✓ | « Je passe du câble au Wi-Fi : seules les couches 1 et 2 changent. » | Les couches du dessus ne voient pas la différence.           |
| ✗ | « La couche 1, c'est l'application, en haut. »                       | On compte depuis le bas : la couche 1 est le câble.          |
| ✗ | « L'adresse IP sert à trouver le bon programme. »                    | L'adresse IP trouve la machine. Le port trouve le programme. |

### CHECKPOINT

À l'oral, 3 minutes. Chacun note sa réponse sur sa feuille, puis on corrige.

1. Donnez les numéros et les noms des couches qui utilisent l'adresse MAC, puis l'adresse IP.
2. Pourquoi découpe-t-on le travail en couches ?

### CAPSULE 2 SUR 4

## Le modèle TCP/IP et la place des protocoles

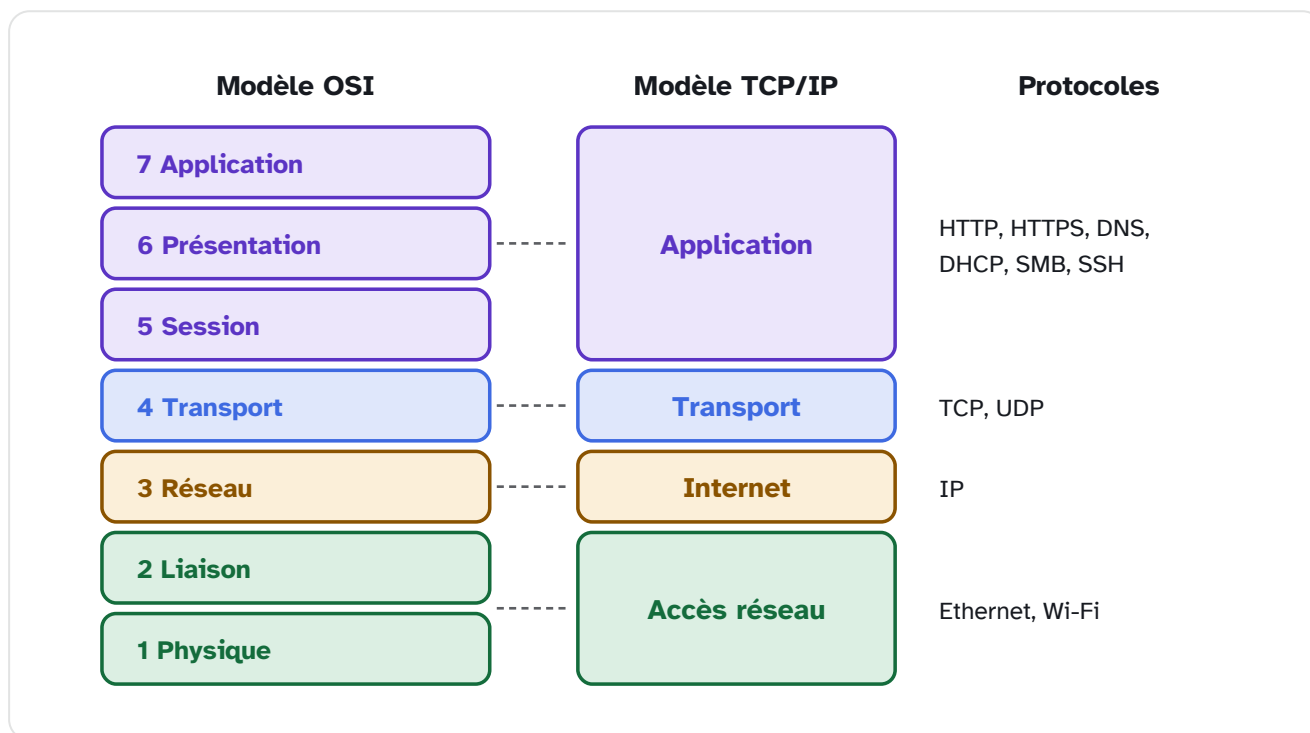
Le modèle OSI sert à expliquer. Mais quel modèle utilisent vraiment les machines d'Internet ?

Elles utilisent le **MODÈLE TCP/IP**, du nom de ses deux protocoles les plus connus. Il a quatre couches seulement. Il regroupe des couches OSI qui travaillent ensemble.

C'est comme deux plans du même lycée. L'un montre chaque salle, l'autre montre seulement les bâtiments. Les deux sont justes : ils ne découpent pas de la même façon.

Voici la correspondance, avec les protocoles les plus courants :

| Couche TCP/IP       | Couches OSI | Protocoles                       | Ce qu'elle apporte                                |
|---------------------|-------------|----------------------------------|---------------------------------------------------|
| <b>Application</b>  | 7, 6 et 5   | HTTP, HTTPS, DNS, DHCP, SMB, SSH | le service : page web, nom, adresse, fichier      |
| <b>Transport</b>    | 4           | TCP, UDP                         | le port, et pour TCP le contrôle de ce qui arrive |
| <b>Internet</b>     | 3           | IP                               | les adresses IP de départ et d'arrivée            |
| <b>Accès réseau</b> | 2 et 1      | Ethernet, Wi-Fi                  | les adresses MAC, et le signal                    |



**Schéma** • Les sept couches OSI regroupées en quatre couches TCP/IP, avec les protocoles de chaque couche

Les deux protocoles de transport n'ont pas le même caractère :

- **TCP** vérifie que tout arrive, dans l'ordre. Il ouvre la conversation avant de parler. On s'en sert pour une page web ou un fichier.
- **UDP** envoie sans vérifier. C'est plus rapide. On s'en sert pour une petite question, comme demander l'adresse d'un site.

#### ON LE FAIT ENSEMBLE — CLASSER QUATRE PROTOCOLES

1. Ethernet relie votre poste au switch, le boîtier qui relie les postes, avec les adresses MAC. Quelle couche TCP/IP ?
2. IP porte l'adresse du poste et celle du serveur. Quelle couche ?
3. TCP porte les ports et vérifie que tout arrive. Quelle couche ?
4. HTTP demande une page web. Quelle couche ? Et dans le modèle OSI ?

Exemple

Pourquoi



DNS, le service qui donne l'adresse d'un site, est rangé en application.

Il rend un service à l'utilisateur, même s'il est petit.

| Exemple                                                       | Pourquoi                                                               |
|---------------------------------------------------------------|------------------------------------------------------------------------|
| ❌ « TCP/IP n'a que quatre couches, donc il oublie le câble. » | Le câble est dans la couche accès réseau, avec la liaison.             |
| ❌ « HTTPS est un protocole de transport, car il protège. »    | HTTPS est du HTTP protégé : il reste en application, au-dessus de TCP. |

### CHECKPOINT

À l'oral, 3 minutes. Chacun note sa réponse sur sa feuille, puis on corrige.

1. Quelles couches OSI la couche application de TCP/IP regroupe-t-elle ?
2. Rangez dans le modèle TCP/IP : UDP, Wi-Fi, SSH.

### CAPSULE 3 SUR 4

## L'encapsulation

Chaque couche a son travail. Mais concrètement, comment les couches se passent-elles les données ?

À l'envoi, chaque couche reçoit les données de la couche du dessus. Elle colle devant un petit bloc d'informations pour faire son travail : un **EN-TÊTE**.

Puis elle passe le tout à la couche du dessous. On appelle ça l'**ENCAPSULATION** : mettre dans une capsule.

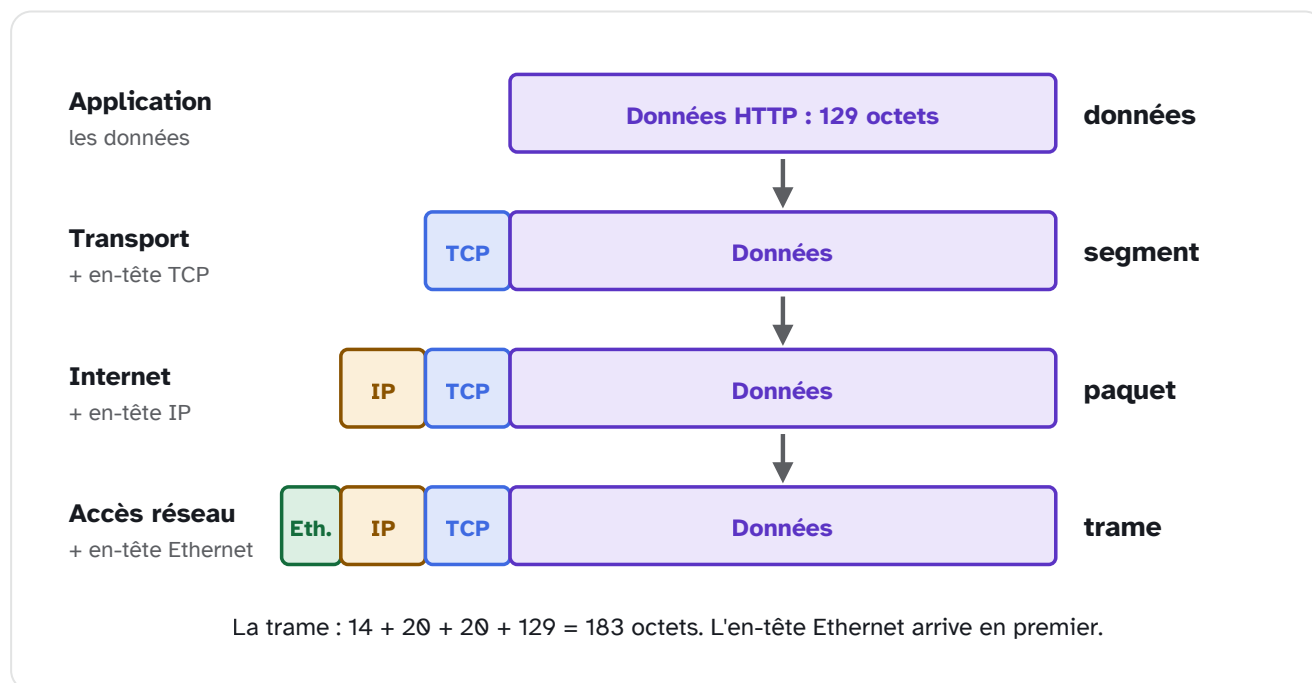
C'est comme des poupées russes. La plus petite est au centre : ce sont les données. Chaque couche ajoute une poupée autour.

Voici l'ordre, quand le poste demande une page web :

- 1 **Application** : le navigateur écrit sa demande HTTP. Ce sont les données.
- 2 **Transport** : TCP ajoute son en-tête, avec le port du client et le port du serveur. Le tout s'appelle un **segment**.
- 3 **Internet** : IP ajoute son en-tête, avec l'adresse IP de départ et celle d'arrivée. Le tout s'appelle un **paquet**.

- 4 **Accès réseau** : Ethernet ajoute son en-tête, avec les adresses MAC. Le tout s'appelle une **trame**. Elle part sur le câble.

À l'arrivée, c'est l'inverse. Chaque couche lit son en-tête, l'enlève, et passe le reste à la couche du dessus.



**Schéma** · Chaque couche ajoute son en-tête devant ce qu'elle reçoit : segment, paquet, trame

Conséquence pratique : dans une trame relevée sur le câble, on lit d'abord l'en-tête Ethernet, puis IP, puis TCP, puis les données. Le dernier ajouté est le premier lu.

#### ON LE FAIT ENSEMBLE — LA DEMANDE DE PAGE DE MDJ-PC-03

Le poste MDJ-PC-03 demande une page à l'imprimante. La trame fait 183 octets. L'en-tête Ethernet fait 14 octets, IP 20 octets, TCP 20 octets.

1. Quel en-tête lit-on en premier dans la trame ?
2. Dans quel en-tête cherche-t-on le port 80 de l'imprimante ?
3. Combien d'octets restent pour la demande HTTP ?

Exemple

Pourquoi



Le poste reçoit la trame : Ethernet enlève son en-tête et passe le paquet à IP.

À l'arrivée, chaque couche enlève son en-tête, de bas en haut.

| Exemple                                      | Pourquoi                                                                      |
|----------------------------------------------|-------------------------------------------------------------------------------|
| ✗ « Le port est dans l'en-tête IP. »         | IP porte les adresses IP. Le port est dans l'en-tête TCP ou UDP.              |
| ✗ « Les données sont au début de la trame. » | Les données sont au centre, à la fin des octets. Les en-têtes passent devant. |

### CHECKPOINT

À l'oral, 3 minutes. Chacun note sa réponse sur sa feuille, puis on corrige.

1. Rangez dans l'ordre d'ajout à l'envoi : en-tête IP, données HTTP, en-tête Ethernet, en-tête TCP.
2. Comment s'appelle ce que fabrique la couche internet ? Et la couche accès réseau ?

### CAPSULE 4 SUR 4

## Reconnaître un protocole dans une capture

On sait maintenant comment une trame est construite. Revenons à Bruno et à son logiciel : comment savoir quel service passe dans une trame ?

Le logiciel de Bruno fait une **CAPTURE** : il enregistre les trames qui passent par une carte réseau, avec tous leurs octets. Le plus connu s'appelle Wireshark, et il est gratuit.

Dans une trame, l'en-tête TCP ou UDP porte deux **PORTS** : un numéro qui désigne le programme, dans la machine. Le serveur écoute sur un port connu, toujours le même pour un service. Le client prend un port au hasard, souvent au-dessus de 49 000.

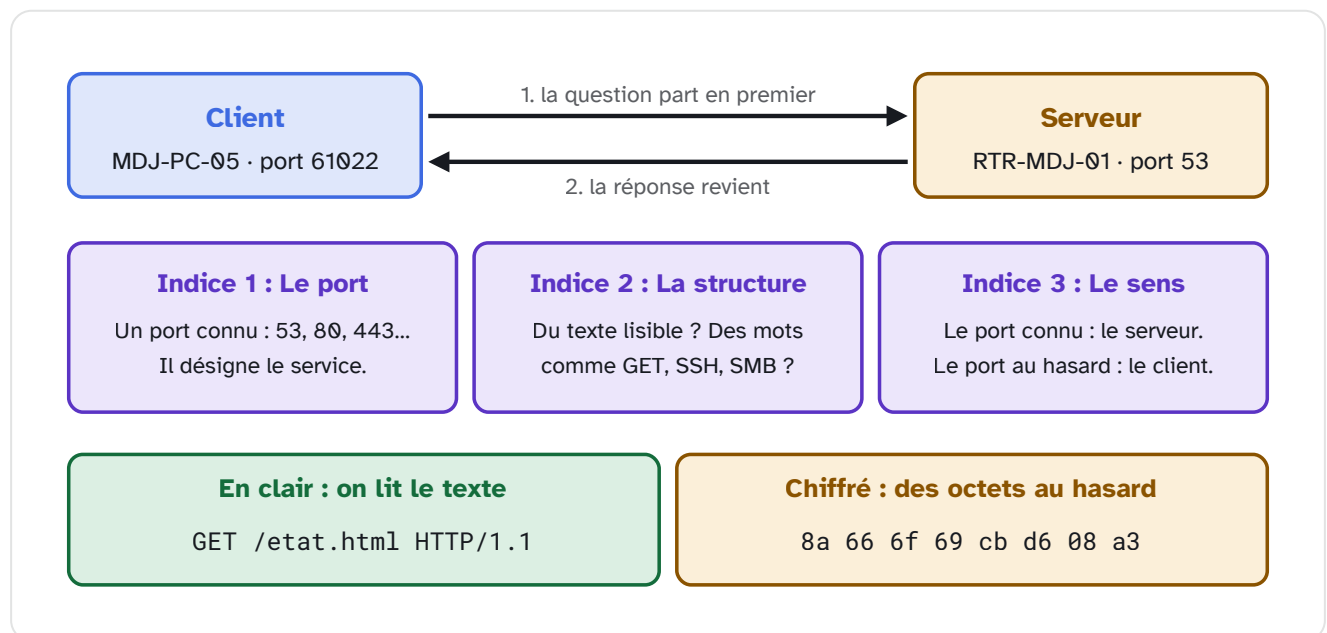
| Port serveur | Transport | Protocole | Service                                        |
|--------------|-----------|-----------|------------------------------------------------|
| 80           | TCP       | HTTP      | pages web                                      |
| 443          | TCP       | HTTPS     | pages web protégées                            |
| 53           | UDP       | DNS       | trouver l'adresse IP d'un nom de site          |
| 67 et 68     | UDP       | DHCP      | donner une adresse IP à une machine qui arrive |
| 22           | TCP       | SSH       | piloter un serveur à distance                  |

| Port serveur | Transport | Protocole | Service                     |
|--------------|-----------|-----------|-----------------------------|
| 445          | TCP       | SMB       | partage de fichiers Windows |

Pour reconnaître un protocole sans en connaître le détail, on cherche trois indices :

- 1 **Le port** : on cherche le port connu dans la table. Il donne une première idée.
- 2 **La structure** : on regarde les données. Du texte lisible ? Des mots comme `GET`, `SSH-2.0` ou `SMB` ? Ou des octets au hasard ?
- 3 **Le sens** : le port connu est celui du serveur, le port au hasard celui du client. Le client parle en premier.

Ce dernier coup d'œil dit aussi si l'échange est **en clair** ou **chiffré**. En clair, on lit le texte. Chiffré, les octets semblent tirés au hasard : seul celui qui a la clé peut les lire.



**Schéma** · Trois indices pour reconnaître un protocole : le port, la structure, le sens ; puis en clair ou chiffré

### ON LE FAIT ENSEMBLE — UNE LIGNE DE CAPTURE

La ligne : 192.168.10.103 vers 192.168.10.20, TCP, du port 52811 vers le port 80. Les données commencent par `GET /etat.html HTTP/1.1`.

1. Quel port est connu ? Que dit la table ?
2. Que montre la structure ?
3. Qui demande ? Qui répond ?
4. En clair ou chiffré ?

#### Exemple

#### Pourquoi

|   |                                                                                   |                                                                                                    |
|---|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| ✓ | Port 22, puis des données qui commencent par <code>SSH-2.0</code> : c'est du SSH. | Deux indices concordent : le port et la structure.                                                 |
| ✗ | « Le port 52811 est le service du client. »                                       | C'est un port pris au hasard. Il ne désigne aucun service connu.                                   |
| ✗ | « Je lis le nom du site dans la capture, donc tout l'échange est en clair. »      | Certaines parties restent visibles, comme le nom du site. Le contenu peut être chiffré quand même. |

### INSTANT HISTOIRE — 2010 : FIRESHEEP, LES COMPTES DES VOISINS DE CAFÉ

En octobre 2010, un développeur, Eric Butler, publie Firesheep, une extension pour le navigateur Firefox.

Dans un café au Wi-Fi ouvert, elle affiche les comptes Facebook ou Twitter des autres clients, en un clic.

Ces sites protégeaient le mot de passe. Mais la suite de la visite passait en clair, avec ce qui prouvait qui vous étiez.

**Qu'est-ce qui aurait empêché Firesheep de voler ces comptes ?**

### **CHECKPOINT**

À l'oral, 3 minutes. Chacun note sa réponse sur sa feuille, puis on corrige.

1. Une ligne montre UDP, du port 61022 vers le port 53. Quel protocole ? Qui est le client ?
2. Donnez deux indices qui disent qu'un échange est chiffré.

## Ce qu'on retient

### **SI VOUS NE DEVEZ RETENIR QUE TROIS CHOSES**

1. Rangez chaque protocole à sa couche : Ethernet en accès réseau, IP en internet, TCP et UDP en transport, HTTP, DNS, SMB et SSH en application.
2. Lisez une trame de l'extérieur vers l'intérieur : Ethernet, puis IP, puis TCP ou UDP, puis les données.
3. Pour nommer un protocole, croisez deux indices : le port connu, et la structure des données.

Revenons à Marc et à Bruno. Bruno a raison sur un point : une capture montre toutes les trames qui passent. Mais il a tort sur l'essentiel.

Vous savez maintenant regarder les données de chaque trame. Si elles sont chiffrées, personne ne peut lire le fichier de Marc, même en voyant passer les trames.

# Le modèle en couches

COURS-U4-S1-02

## L'essentiel

- **Protocole** : les règles communes de deux machines qui se parlent. Chacun fait un seul travail.
- **Modèle OSI** : sept couches, de 1, physique, à 7, application. Il sert à expliquer.
- **Modèle TCP/IP** : quatre couches, celles qu'utilise Internet : accès réseau, internet, transport, application.
- **Encapsulation** : à l'envoi, chaque couche ajoute son en-tête devant les données. À l'arrivée, chaque couche enlève le sien.
- **Reconnaître un protocole** : le port connu, la structure des données, le sens de l'échange.

## Les mots à connaître

| Mot                           | En clair                                                                           |
|-------------------------------|------------------------------------------------------------------------------------|
| <b>Couche</b>                 | Un étage du réseau, avec un seul travail.                                          |
| <b>En-tête</b>                | Le bloc d'informations qu'une couche colle devant les données.                     |
| <b>Segment, paquet, trame</b> | Ce que fabriquent le transport, la couche internet, l'accès réseau.                |
| <b>Port</b>                   | Le numéro du programme dans la machine. Connu côté serveur, au hasard côté client. |
| <b>Capture</b>                | L'enregistrement des trames qui passent par une carte réseau.                      |
| <b>En clair</b>               | Le texte des données se lit dans la capture.                                       |
| <b>Chiffré</b>                | Les octets semblent tirés au hasard. Seul celui qui a la clé peut les lire.        |

## Les deux modèles

| TCP/IP       | OSI     | Protocoles                       | L'en-tête apporte |
|--------------|---------|----------------------------------|-------------------|
| Application  | 7, 6, 5 | HTTP, HTTPS, DNS, DHCP, SMB, SSH | le service        |
| Transport    | 4       | TCP, UDP                         | les ports         |
| Internet     | 3       | IP                               | les adresses IP   |
| Accès réseau | 2, 1    | Ethernet, Wi-Fi                  | les adresses MAC  |

## Les ports à connaître

| Port     | Protocole | Port          | Protocole |
|----------|-----------|---------------|-----------|
| 80, TCP  | HTTP      | 22, TCP       | SSH       |
| 443, TCP | HTTPS     | 445, TCP      | SMB       |
| 53, UDP  | DNS       | 67 et 68, UDP | DHCP      |

## La méthode

1. **Repérez** le transport, TCP ou UDP, et les deux ports.
2. **Cherchez** le port connu dans la table : c'est le serveur.
3. **Regardez** les données : texte lisible, mot connu, ou octets au hasard ?
4. **Nommez** le protocole avec deux indices qui concordent.
5. **Dites** qui demande et qui répond.

## Un exemple concret

Une ligne de capture.

```
192.168.10.105 → 192.168.10.1    UDP    61022 → 53
Données : ...www.federation-jeux.example...
```

| Étape | Ce qu'on fait      | Résultat                                    |
|-------|--------------------|---------------------------------------------|
| 1     | Transport et ports | UDP, 61022 et 53                            |
| 2     | Port connu         | 53 : DNS, côté serveur                      |
| 3     | Données            | un nom de site lisible                      |
| 4     | Deux indices       | port 53 et nom de site : DNS                |
| 5     | Le sens            | 192.168.10.105 demande, 192.168.10.1 répond |

**Conclusion :** c'est une demande DNS, en clair. Le poste demande l'adresse IP de `www.federation-jeux.example`.

## Les pièges

- **✗ Compter les couches depuis le haut** → la couche 1 est en bas : le câble.
- **✗ Chercher le port dans l'en-tête IP** → il est dans l'en-tête TCP ou UDP.
- **✗ Prendre le port du client pour le service** → le service est sur le port connu, côté serveur.
- **✗ Croire le port seul** → un service peut changer de port. Confirmez avec la structure.

# Lire une capture couche par couche

LABO-U4-S1-02

**Contexte** — Vous êtes prestataire bénévole chez MaisonDesJeux, une association de jeux de société. L'équipe pédagogique a enregistré six échanges sur la maquette du futur réseau de l'association.

Marc Oliveira, le trésorier, veut savoir si quelqu'un peut lire le fichier des adhérents sur le réseau. Bruno Sanchez, un bénévole, affirme que « tout passe en clair ». Vous allez lire les six captures pour répondre.

## RÈGLE IA DE LA SÉANCE

Scénario 1 : pas d'IA pendant ce labo. Vous avez les annexes, l'aide de Wireshark et le web classique.

Si vous avez un doute, demandez au formateur ou à votre binôme.

**À la fin de ce labo, vous saurez** lire une trame couche par couche et nommer un protocole avec deux indices. Vous saurez aussi dire si un échange est en clair ou chiffré.

**Comment ça se valide** — à chaque point de contrôle, vous montrez votre résultat au formateur. Il valide, puis vous passez à la suite.

## Les mots de la séance

| Mot       | Ce que ça veut dire                                                                                |
|-----------|----------------------------------------------------------------------------------------------------|
| Capture   | L'enregistrement des trames qui passent par une carte réseau. Ici, un fichier <code>.pcap</code> . |
| Trame     | Ce qui circule sur le câble : les données et tous leurs en-têtes.                                  |
| En-tête   | Le bloc d'informations qu'une couche colle devant les données.                                     |
| Port      | Le numéro du programme dans la machine. Connu côté serveur, au hasard côté client.                 |
| Wireshark | Le logiciel gratuit qui ouvre et affiche les captures.                                             |
| En clair  | On lit le texte des données dans la capture.                                                       |
| Chiffré   | Les octets semblent tirés au hasard. Seul celui qui a la clé peut les lire.                        |

## Ce dont vous avez besoin

- Votre poste Windows, avec Wireshark.
- [Annexe A1](#) : les six captures à ouvrir dans Wireshark.
- [Annexe A2](#) : extrait du guide de Wireshark.
- [Annexe A3](#) : la trame à annoter, imprimée, et deux stylos de couleurs différentes.
- [Annexe A4](#) : les six captures imprimées.
- [Annexe A5](#) : la table des ports usuels.
- [Annexe A6](#) : les machines de la maquette MaisonDesJeux.

**En binôme** — le pilote manipule Wireshark, le lecteur lit les consignes et les annexes à voix haute, et vérifie chaque résultat. On inverse à la partie C.

## Le plan du labo

| Partie                                    | Étapes  | Ce que vous obtenez                              | Temps  |
|-------------------------------------------|---------|--------------------------------------------------|--------|
| A — Voir les couches d'une trame          | 1 à 6   | la liste des couches de deux trames              | 30 min |
| B — Annoter une trame                     | 7 à 12  | la trame de l' <a href="#">annexe A3</a> annotée | 25 min |
| C — Nommer six protocoles                 | 13 à 16 | la grille des six protocoles, avec deux indices  | 35 min |
| D — En clair ou chiffré, et qui demande ? | 17 à 22 | la réponse à Marc Oliveira                       | 30 min |

## PARTIE A — Voir les couches d'une trame

*Étapes 1 à 6, environ 30 min. À la fin, vous savez dans quelle couche range chaque ligne du détail de Wireshark.*

*Besoin d'un rappel ? Revoir le cours : « [Le modèle TCP/IP et la place des protocoles](#) ».*

## 01 Récupérer les captures

- ☐ Téléchargez le fichier `A1_captures_mdj.zip` depuis l'[annexe A1](#).
- ☐ Faites un clic droit sur le zip, puis « Extraire tout ». Choisissez le dossier « Documents », puis cliquez sur « Extraire ».

**Vous devez voir :** un dossier `Documents\A1_captures_mdj` avec six fichiers, de `capture1.pcap` à `capture6.pcap`.

## 02 Ouvrir la capture 1

- ☐ Lancez Wireshark depuis le menu Démarrer.
- ☐ Ouvrez le menu Fichier > Ouvrir (File > Open), puis choisissez `capture1.pcap`.

**Vous devez voir :** 8 lignes dans la fenêtre. Les lignes 4 et 5 portent le mot HTTP dans la colonne Protocol.

**Si vous voyez** une fenêtre vide avec la liste des cartes réseau : la capture n'est pas ouverte. Refaites Fichier > Ouvrir.

## 03 Trouver dans la doc

La fenêtre de Wireshark est découpée en trois panneaux. L'[annexe A2](#) explique chacun.

- ☐ Ouvrez l'[annexe A2](#), partie 2.

☐ Trouvez le nom du panneau qui montre les en-têtes, couche par couche. Notez-le :

\_\_\_\_\_

☐ Trouvez ce que Wireshark écrit à la place d'un octet qui n'est pas une lettre, à droite du panneau 3. Notez-le : \_\_\_\_\_

**Vous devez avoir :** le nom du panneau du milieu, et un seul caractère.

## 04 Lire le détail de la trame 4

☐ Cliquez sur la ligne 4 de la liste, celle qui contient `GET /etat.html`.

☐ Dans le panneau du détail, recopiez le début de chaque ligne, de haut en bas. Arrêtez-vous à la première virgule, s'il y en a une :

| N° | Début de la ligne, dans le panneau du détail |
|----|----------------------------------------------|
|----|----------------------------------------------|

|   |                 |
|---|-----------------|
| 1 | Frame 4: Packet |
|---|-----------------|

|   |       |
|---|-------|
| 2 | _____ |
|---|-------|

|   |       |
|---|-------|
| 3 | _____ |
|---|-------|

|   |       |
|---|-------|
| 4 | _____ |
|---|-------|

|   |       |
|---|-------|
| 5 | _____ |
|---|-------|

**Vous devez avoir :** cinq lignes. La ligne 2 commence par « Ethernet II ».

**Si vous voyez** des lignes ouvertes avec beaucoup de détails : cliquez sur la petite flèche à gauche pour les refermer.

## 05 Ranger chaque ligne dans les modèles

La première ligne, Frame, n'est pas une couche. Wireshark y résume la trame : sa taille, son heure. Les quatre autres lignes sont des en-têtes de protocoles.

- ☐ Remplissez la grille avec le tableau des deux modèles du cours. La première ligne est faite pour vous.

| Ligne du détail | Protocole | Couche TCP/IP | Couche OSI |
|-----------------|-----------|---------------|------------|
| Ethernet II     | Ethernet  | Accès réseau  | 2, liaison |
|                 |           |               |            |
|                 |           |               |            |
|                 |           |               |            |

**Vous devez avoir :** quatre couches TCP/IP différentes, de l'accès réseau en haut à l'application en bas.

**Pourquoi ?** Le modèle OSI a sept couches. Pourquoi Wireshark n'affiche-t-il que quatre lignes de protocoles ?

Votre réponse :

## 06 Comparer avec la trame 1

- ☐ Cliquez sur la ligne 1 de la liste. Elle porte **[SYN]** dans la colonne Info.

☐ Comptez les lignes du panneau du détail. Notez leur nombre :

\_\_\_\_\_

**Vous devez voir :** quatre lignes, sans aucune ligne HTTP.

**Pourquoi ?** La trame 1 ouvre la conversation TCP. Pourquoi n'a-t-elle pas de ligne HTTP ?

Votre réponse :

#### POINT DE CONTRÔLE A

☐ Montrez au formateur votre grille de l'étape 5 et votre réponse de l'étape 6.

☐ Le formateur a validé. Passez à la partie B.

## PARTIE B — Annoter une trame couche par couche

Étapes 7 à 12, environ 25 min. À la fin, la trame de l'[annexe A3](#) est découpée en couches, au stylo.

Besoin d'un rappel ? Revoir le cours : « [L'encapsulation](#) ».

L'[annexe A3](#) est la trame 4 de la capture 1, imprimée octet par octet. Vous allez trouver où commence et où finit chaque en-tête.

## 07 Surligner les octets d'une couche

L'[annexe A2](#), partie 3, explique comment Wireshark montre les octets d'une couche.

---

☐ Revenez sur la trame 4. Dans le panneau du détail, cliquez sur la ligne « Ethernet II ».

---

☐ Regardez le panneau des octets, en bas.

---

**Vous devez voir :** les 14 premiers octets surlignés, de `02 4d` à `08 00`.

## 08 Encadrer l'en-tête Ethernet

---

☐ Sur l'[annexe A3](#), encadrez au stylo les octets 0 à 13.

---

☐ Écrivez à côté du cadre : « Ethernet, accès réseau ».

---

**Vous devez avoir :** un cadre de 14 octets, qui finit deux octets avant la fin de la première ligne.

## 09 Encadrer l'en-tête IP

---

☐ Dans Wireshark, cliquez sur la ligne « Internet Protocol Version 4 ». Repérez les octets surlignés.

---

☐ Sur l'[annexe A3](#), encadrez ces octets avec l'autre couleur. Écrivez « IP, internet ».

---

**Vous devez avoir :** un cadre qui commence à l'octet 14, juste après Ethernet.

## 10 Encadrer l'en-tête TCP et les données

☐ Faites de même pour la ligne « Transmission Control Protocol ». Écrivez « TCP, transport ».

☐ Faites de même pour la ligne « Hypertext Transfer Protocol ». Écrivez « HTTP, application ».

**Vous devez avoir :** quatre cadres qui se suivent, sans trou, jusqu'au dernier octet de la trame.

## 11 Remplir la grille de la trame

Chaque en-tête porte les informations de sa couche. Wireshark les montre si vous cliquez sur la petite flèche à gauche de la ligne.

☐ Remplissez la grille. La première ligne est faite pour vous. Taille = dernier octet – premier octet + 1.

| Couche            | Octets, du premier au dernier | Taille               | Deux informations de l'en-tête                                  |
|-------------------|-------------------------------|----------------------|-----------------------------------------------------------------|
| Ethernet          | 0 à 13                        | 14 octets            | MAC destination 02:4d:44:4a:00:20, MAC source 02:4d:44:4a:01:03 |
| IP                | <input type="text"/>          | <input type="text"/> | <input type="text"/>                                            |
| TCP               | <input type="text"/>          | <input type="text"/> | <input type="text"/>                                            |
| HTTP, les données | <input type="text"/>          | <input type="text"/> | la page demandée : <input type="text"/>                         |

**Vous devez avoir :** les adresses IP dans la ligne IP, les ports dans la ligne TCP.

## 12 Vérifier le total

- ☐ Additionnez les quatre tailles de votre grille :

..... + ..... + ..... + ..... = ..... octets

- ☐ Comparez avec la taille annoncée sur la ligne « Frame 4 » du détail.

**Vous devez avoir :** le même nombre des deux côtés.

**Pourquoi ?** Ethernet est la couche du bas. Pourquoi ses octets sont-ils les premiers de la trame ?

Votre réponse :

### POINT DE CONTRÔLE B

- ☐ Montrez au formateur l'[annexe A3](#) annotée et votre grille de l'étape 11.

- ☐ Le formateur a validé. Inversez les rôles, puis passez à la partie C.

## PARTIE C — Nommer le protocole des six captures

Étapes 13 à 16, environ 35 min. À la fin, vous avez nommé chaque protocole, avec deux indices.

Besoin d'un rappel ? Revoir le cours : [« Reconnaître un protocole dans une capture »](#).

Wireshark écrit le nom du protocole dans la colonne Protocol. Ici, vous le trouvez d'abord sans lui, sur papier. L'[annexe A4](#) donne les six captures imprimées, sans le nom du protocole.

### 13 Lire l'exemple de la capture 1

☐ Ouvrez l'[annexe A4](#) à la capture 1, et l'[annexe A5](#), la table des ports.

☐ Lisez la ligne d'exemple de la grille ci-dessous. Retrouvez chaque indice dans l'[annexe A4](#).

| Capture | Transport | Port du serveur | Protocole | Indice 1 : le port                                       | Indice 2 : la structure                                 |
|---------|-----------|-----------------|-----------|----------------------------------------------------------|---------------------------------------------------------|
| 1       | TCP       | 80              | HTTP      | 80 est le port de HTTP dans l' <a href="#">annexe A5</a> | les données sont du texte lisible, qui commence par GET |
| 2       | _____     | _____           | _____     | _____                                                    | _____                                                   |
| 3       | _____     | _____           | _____     | _____                                                    | _____                                                   |
| 4       | _____     | _____           | _____     | _____                                                    | _____                                                   |
| 5       | _____     | _____           | _____     | _____                                                    | _____                                                   |
| 6       | _____     | _____           | _____     | _____                                                    | _____                                                   |

**Vous devez voir :** dans l'[annexe A4](#), le port 80 dans la colonne Ports, et GET /etat.html dans la colonne Début des données.

## 14 Nommer les captures 2 et 3

- ☐ Remplissez les lignes 2 et 3 de la grille, avec l'[annexe A4](#) et l'[annexe A5](#).

**Vous devez avoir :** un protocole TCP et un protocole UDP. La capture 3 contient un nom de site lisible.

## 15 Nommer les captures 4, 5 et 6

- ☐ Remplissez les lignes 4, 5 et 6 de la grille.

**Vous devez avoir :** pour la capture 6, deux ports connus. Ils appartiennent au même protocole.

**Si vous voyez** des données sans aucun texte lisible : c'est aussi un indice de structure.  
Écrivez « aucun texte lisible ».

**Pourquoi ?** Bruno pourrait régler un serveur web sur le port 8080 au lieu de 80. Pourquoi le port seul ne suffit-il pas à nommer un protocole ?

Votre réponse :

## 16 Vérifier dans Wireshark

- ☐ Ouvrez une à une les captures 2 à 6 dans Wireshark, avec Fichier > Ouvrir.
- ☐ Pour chacune, lisez la colonne Protocol. Notez les cinq noms affichés :  
\_\_\_\_\_
- ☐ Comparez avec votre grille et avec la dernière colonne de l'[annexe A5](#). Corrigez votre grille au besoin.

**Vous devez avoir :** le même protocole que dans votre grille. Lisez les lignes qui ont des données.

Pour la capture 2, Wireshark affiche `TLSv1.2` ou `TLSv1.3` : c'est le TLS de HTTPS.

### POINT DE CONTRÔLE C

- ☐ Montrez au formateur votre grille complète. Expliquez-lui les deux indices de la capture qu'il choisit.
- ☐ Le formateur a validé. Passez à la partie D.

## PARTIE D — En clair ou chiffré, et qui demande ?

Étapes 17 à 22, environ 30 min. À la fin, vous pouvez répondre à Marc Oliveira.

Wireshark peut afficher les données d'une conversation entière, sans les en-têtes. C'est la fonction « Suivre le flux », expliquée dans l'[annexe A2](#), partie 4.

## 17 Suivre le flux de la capture 1

- ☐ Ouvrez `capture1.pcap`. Faites un clic droit sur la ligne 4, puis Suivre > Flux TCP (Follow > TCP Stream).

- ☐ Lisez le texte. Notez le niveau du toner noir de l'imprimante :  
\_\_\_\_\_

**Vous devez voir :** la demande du poste en rouge, la page de l'imprimante en bleu, en texte lisible.

**Si vous voyez** une seule partie des trames après avoir fermé la fenêtre : Wireshark a laissé un filtre. Effacez la barre du filtre, puis appuyez sur Entrée.

## 18 Suivre le flux de la capture 2

- ☐ Ouvrez `capture2.pcap`. Faites un clic droit sur la ligne 4, puis Suivre > Flux TCP.

- ☐ Cherchez dans le texte un mot lisible qui donne le nom du site. Notez-le :  
\_\_\_\_\_

**Vous devez voir :** surtout des points et des caractères au hasard. Au début, quelques mots se lisent : un nom de site, puis `h2` et `http/1.1`.

**Pourquoi ?** Un observateur lit le nom du site, mais pas la page. Qu'est-ce qui reste caché ?

Votre réponse :

## 19 Suivre le flux de la capture 4

C'est l'échange qui inquiète Marc : un poste ouvre le fichier des adhérents sur le serveur.

☐ Ouvrez `capture4.pcap`. Faites un clic droit sur la ligne 1, puis Suivre > Flux TCP.

☐ Cherchez un nom de fichier, un nom d'adhérent ou un montant.

**Vous devez voir :** les trois lettres `SMB` au début de chaque message, puis des caractères au hasard. Aucun nom lisible.

## 20 Classer les six échanges

☐ Remplissez la grille avec vos étapes 17 à 19 et l'[annexe A4](#). La première ligne est faite pour vous.

Un mot lisible au début, comme `SSH-2.0` ou `SMB`, ne suffit pas : regardez les trames suivantes. Pour la capture 6, cherchez un nom de machine dans les octets du détail.

| Capture | En clair ou chiffré ? | À quoi je le vois                                                     |
|---------|-----------------------|-----------------------------------------------------------------------|
| 1       | en clair              | Je lis la demande <code>GET /etat.html</code> et le texte de la page. |
| 2       | _____                 | _____                                                                 |

| Capture | En clair ou chiffré ? | À quoi je le vois |
|---------|-----------------------|-------------------|
| 3       | _____                 | _____             |
| 4       | _____                 | _____             |
| 5       | _____                 | _____             |
| 6       | _____                 | _____             |

**Vous devez avoir :** trois échanges en clair et trois échanges chiffrés.

## 21 Dire qui demande et qui répond

Le client a un port au hasard. Le serveur a le port connu. Pour DHCP, le client a le port 68, le serveur le port 67.

L'[annexe A6](#) donne le nom de chaque machine à partir de son adresse.

☐

Remplissez la grille avec l'[annexe A4](#) et l'[annexe A6](#). La première ligne est faite pour vous.

| Capture | Qui demande : le client      | Qui répond : le serveur      | Comment je le sais                                                  |
|---------|------------------------------|------------------------------|---------------------------------------------------------------------|
| 1       | MDJ-PC-03,<br>192.168.10.103 | IMP-MDJ-01,<br>192.168.10.20 | MDJ-PC-03 a le port 52811, au hasard.<br>L'imprimante a le port 80. |
| 2       | _____                        | _____                        | _____                                                               |
| 3       | _____                        | _____                        | _____                                                               |
| 4       | _____                        | _____                        | _____                                                               |
| 5       | _____                        | _____                        | _____                                                               |
| 6       | _____                        | _____                        | _____                                                               |

**Vous devez avoir :** quatre échanges où répond **SRV-MDJ-01** ou **RTR-MDJ-01** , et un seul serveur sur Internet.

**Si vous voyez** l'adresse MAC du routeur dans la capture 2 alors que le serveur est sur Internet : c'est normal. Le poste confie la trame au routeur, qui la fait sortir.

## 22 Expliquer la capture 6 et répondre à Marc

- ☐ Regardez la source de la trame 1 de la capture 6 : `0.0.0.0` . Relisez les adresses spéciales, à la fin de l'[annexe A6](#).

**Pourquoi ?** Pourquoi le portable de Léa Fournier envoie-t-il depuis l'adresse `0.0.0.0`, vers tout le réseau ?

Votre réponse :

- ☐ Écrivez en deux phrases la réponse à Marc Oliveira. Citez la capture et l'indice qui le prouve.

Votre réponse :

**Vous devez avoir :** une réponse oui ou non, la capture qui le prouve, et un indice lu dans les octets.

#### POINT DE CONTRÔLE D

- ☐ Montrez au formateur vos grilles des étapes 20 et 21, et votre réponse à Marc.
- ☐ Le formateur a validé. Le labo est terminé.

## POUR ALLER PLUS LOIN

10 à 15 min, si vous avez fini avant la fin de la séance.

Marc utilise le poste de la capture 2.

Bruno Sanchez dit : « La banque, c'est chiffré. Donc personne ne sait que Marc est allé sur le site de la banque. »

- ☐ Ouvrez `capture2.pcap` . Pour chaque information, dites si un observateur du réseau la voit quand même :

| Information                          | Visible ? oui ou non | Où je la lis |
|--------------------------------------|----------------------|--------------|
| L'adresse IP du poste de Marc        | _____                | _____        |
| L'adresse IP du serveur de la banque | _____                | _____        |
| Le nom du site                       | _____                | _____        |
| Le montant du solde du compte        | _____                | _____        |
| L'heure de la visite                 | _____                | _____        |

- ☐ Écrivez en une phrase l'erreur de Bruno : \_\_\_\_\_

## CE QU'IL FAUT RETENIR

1. Une trame se lit de l'extérieur vers l'intérieur : Ethernet, IP, TCP ou UDP, puis les données de l'application.
2. Un protocole se nomme avec deux indices qui concordent : le port connu, et la structure des données.
3. Chiffré ne veut pas dire invisible : on voit qui parle à qui, et quand. Mais on ne lit pas le contenu.

# Tous les documents du module

## A1 — Les six captures à ouvrir dans Wireshark

↓ Télécharger A1\_captures\_mdj.zip

Source : captures relevées sur la maquette du réseau cible de MaisonDesJeux, d'après le document de scénario, § 5 « Captures de référence ». Fichier [A1\\_captures\\_mdj.zip](#) à télécharger ci-dessus. Utilisée dans : LABO étapes 1, 2, 16 à 19.

L'équipe pédagogique a monté la maquette du réseau que l'association aura en fin d'année. Elle a enregistré six échanges, le mercredi 16 septembre 2026, entre 14 h 30 et 16 h 05.

Le fichier [A1\\_captures\\_mdj.zip](#) contient six captures, au format `.pcap` :

| Fichier       | Nombre de trames | Heure du relevé |
|---------------|------------------|-----------------|
| capture1.pcap | 8                | 14 h 32         |
| capture2.pcap | 12               | 14 h 40         |
| capture3.pcap | 2                | 14 h 45         |
| capture4.pcap | 7                | 14 h 51         |
| capture5.pcap | 19               | 15 h 03         |
| capture6.pcap | 4                | 16 h 02         |

### Le mode d'emploi

1. Téléchargez le zip, puis extrayez-le dans « Documents » : clic droit, « Extraire tout ».
2. Lancez Wireshark, puis ouvrez une capture avec Fichier > Ouvrir.
3. Pour repartir de zéro, supprimez le dossier `A1_captures_mdj` et extrayez de nouveau le zip.

Ces captures ne contiennent aucune vraie donnée : les adresses externes sont des adresses réservées aux exemples.

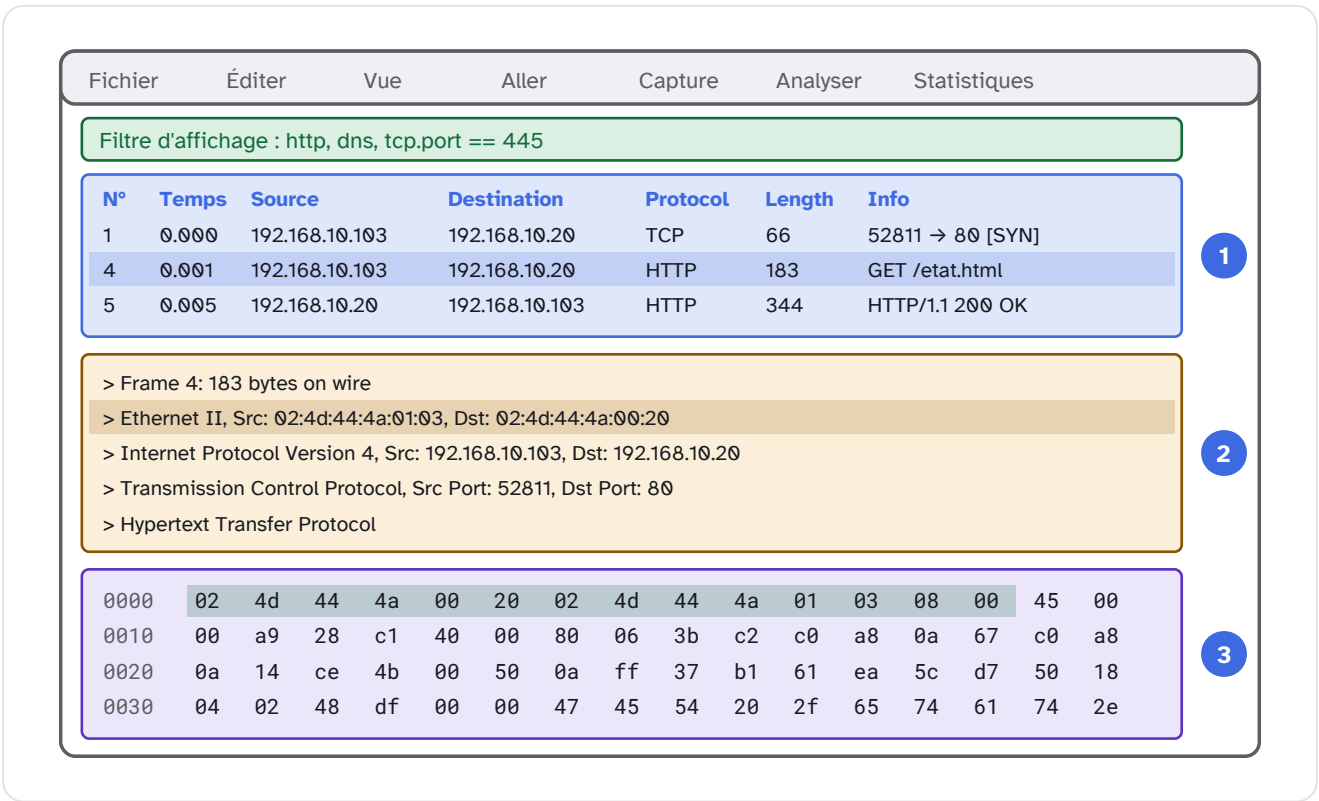
# A2 — Extrait du guide de Wireshark

Source : guide de l'utilisateur de Wireshark, [wireshark.org](https://www.wireshark.org), chapitres 3 et 7, traduits et raccourcis pour le module. Utilisée dans : LABO étapes 3, 7 et 17.

## Partie 1 — Ouvrir une capture

Menu Fichier > Ouvrir (File > Open), puis choisissez le fichier. Un double clic sur un fichier `.pcap` l'ouvre aussi, si Wireshark est le programme associé.

## Partie 2 — Les trois panneaux



**Schéma** • La fenêtre de Wireshark : 1 la liste des trames, 2 le détail couche par couche, 3 les octets

| N° | Nom du panneau                    | Ce qu'il montre                                                                                                                                               |
|----|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Liste des paquets (Packet List)   | Une ligne par trame : numéro, temps, source, destination, protocole, longueur, résumé.                                                                        |
| 2  | Détail du paquet (Packet Details) | Les en-têtes de la trame choisie, une ligne par protocole. La couche la plus basse est en haut. La première ligne, Frame, est un résumé ajouté par Wireshark. |

| N° | Nom du panneau                  | Ce qu'il montre                                                                                                                                                                                                                                                                                                                  |
|----|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3  | Octets du paquet (Packet Bytes) | Tous les octets de la trame, en hexadécimal, 16 par ligne. À gauche, le numéro du premier octet de la ligne. À droite, les octets lisibles comme des lettres ; un point  remplace un octet qui n'est ni une lettre, ni un chiffre, ni un signe. |

Wireshark écrit « paquet » pour chaque ligne, même quand c'est une trame Ethernet.

Dans le panneau 2, une petite flèche à gauche d'une ligne l'ouvre : on voit alors chaque champ de l'en-tête.

### Partie 3 — Voir les octets d'un en-tête

Cliquez sur une ligne du panneau 2. Wireshark surligne ses octets dans le panneau 3. Cliquez sur un champ, une adresse par exemple : seuls ses octets sont surlignés.

### Partie 4 — Suivre un flux

Faites un clic droit sur une trame, puis Suivre > Flux TCP (Follow > TCP Stream). Pour un échange en UDP, choisissez UDP Stream : ce choix n'est pas traduit. Le même choix existe dans le menu Analyser.

Une fenêtre affiche les données de l'application de toute la conversation, sans les en-têtes :

- en rouge, ce que le client envoie ;
- en bleu, ce que le serveur répond ;
- un point  remplace un octet qui n'est pas une lettre.

En fermant cette fenêtre, Wireshark garde un filtre dans la barre du haut, par exemple `tcp.stream eq 0`. Effacez-le pour revoir toutes les trames.

### Partie 5 — Filtrer l'affichage

Tapez un filtre dans la barre du filtre, en haut, puis appuyez sur Entrée. Seules les trames qui y répondent restent affichées.

| Filtre            | Ce qui reste affiché               |
|-------------------|------------------------------------|
| <code>http</code> | les trames qui contiennent du HTTP |
| <code>dns</code>  | les trames qui contiennent du DNS  |

tcp.port == 445

les trames TCP dont un des deux ports vaut 445

Une barre rouge signale un filtre mal écrit.

## A3 — La trame à annoter

Source : trame 4 de `capture1.pcap`, [annexe A1](#). Utilisée dans : LABO étapes 8 à 12. À imprimer, une par élève.

Le poste `MDJ-PC-03` demande une page à l'imprimante `IMP-MDJ-01`. La trame fait 183 octets.

### Comment lire ce relevé

- Chaque ligne montre 16 octets.
- La première colonne donne le numéro du premier octet de la ligne, en décimal. Le premier octet porte le numéro 0.
- Après le signe `.`, le même numéro en hexadécimal, comme dans Wireshark : `0010` est l'octet 16.
- Pour trouver le numéro d'un octet, partez du début de la ligne et comptez : 0, 1, 2...
- La dernière colonne montre les octets lisibles comme des lettres. Un point remplace un octet qui n'est ni une lettre, ni un chiffre, ni un signe.

| N°  | hexa   | octets                                          | lettres          |
|-----|--------|-------------------------------------------------|------------------|
| 0   | · 0000 | 02 4d 44 4a 00 20 02 4d 44 4a 01 03 08 00 45 00 | .MDJ. .MDJ....E. |
| 16  | · 0010 | 00 a9 28 c1 40 00 80 06 3b c2 c0 a8 0a 67 c0 a8 | ..(.@...;....g.. |
| 32  | · 0020 | 0a 14 ce 4b 00 50 0a ff 37 b1 61 ea 5c d7 50 18 | ...K.P..7.a.\.P. |
| 48  | · 0030 | 04 02 48 df 00 00 47 45 54 20 2f 65 74 61 74 2e | ..H...GET /etat. |
| 64  | · 0040 | 68 74 6d 6c 20 48 54 54 50 2f 31 2e 31 0d 0a 48 | html HTTP/1.1..H |
| 80  | · 0050 | 6f 73 74 3a 20 31 39 32 2e 31 36 38 2e 31 30 2e | ost: 192.168.10. |
| 96  | · 0060 | 32 30 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 | 20..User-Agent:  |
| 112 | · 0070 | 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 57 69 6e | Mozilla/5.0 (Win |
| 128 | · 0080 | 64 6f 77 73 20 4e 54 20 31 30 2e 30 29 0d 0a 41 | dows NT 10.0)..A |
| 144 | · 0090 | 63 63 65 70 74 3a 20 74 65 78 74 2f 68 74 6d 6c | ccept: text/html |
| 160 | · 00a0 | 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c | ..Connection: cl |
| 176 | · 00b0 | 6f 73 65 0d 0a 0d 0a                            | ose....          |

Sous le relevé, gardez de la place pour vos notes :

| Couche | Octets, du premier au dernier | Ce que j'ai encadré |
|--------|-------------------------------|---------------------|
|        |                               |                     |
|        |                               |                     |
|        |                               |                     |
|        |                               |                     |

## A4 — Les six captures imprimées

*Source : transcription des six captures de l'[annexe A1](#), faite par l'équipe pédagogique. Le nom du protocole d'application est masqué. Utilisée dans : LABO étapes 13 à 15, 20 et 21. À imprimer, une par binôme.*

### Comment lire la liste des trames

- **N°** : le numéro de la trame. **Temps** : les secondes depuis la première trame.
- **Transport** : TCP ou UDP. **Ports** : port de la source, puis port de la destination.
- **Lg** : la longueur de la trame, en octets.
- **Début des données** : les 20 premiers octets des données de l'application, lus comme des lettres. Un point remplace un octet qui n'est pas une lettre. Entre crochets, une trame TCP

sans données : seulement ses drapeaux. [SYN] ouvre la conversation, [ACK] dit « bien reçu », [FIN] la ferme.

**Comment lire le détail d'une trame** : les en-têtes Ethernet, IP et TCP ou UDP, avec leurs champs utiles. Puis les octets des données, comme dans le panneau des octets de Wireshark. La première colonne y est en hexadécimal. Le signe ... marque des octets non imprimés.

## Capture 1

Relevée à 14 h 32

| N°                   | Temps | Source         | Destination    | Transport | Ports      | Lg  |
|----------------------|-------|----------------|----------------|-----------|------------|-----|
| Début des données    |       |                |                |           |            |     |
| ---                  | ----- | -----          | -----          | -----     | -----      | --- |
| 1                    | 0.000 | 192.168.10.103 | 192.168.10.20  | TCP       | 52811 → 80 | 66  |
| [SYN]                |       |                |                |           |            |     |
| 2                    | 0.001 | 192.168.10.20  | 192.168.10.103 | TCP       | 80 → 52811 | 66  |
| [SYN, ACK]           |       |                |                |           |            |     |
| 3                    | 0.001 | 192.168.10.103 | 192.168.10.20  | TCP       | 52811 → 80 | 54  |
| [ACK]                |       |                |                |           |            |     |
| 4                    | 0.001 | 192.168.10.103 | 192.168.10.20  | TCP       | 52811 → 80 | 183 |
| GET /etat.html HTTP/ |       |                |                |           |            |     |
| 5                    | 0.005 | 192.168.10.20  | 192.168.10.103 | TCP       | 80 → 52811 | 344 |
| HTTP/1.1 200 OK..Ser |       |                |                |           |            |     |
| 6                    | 0.006 | 192.168.10.103 | 192.168.10.20  | TCP       | 52811 → 80 | 54  |
| [ACK]                |       |                |                |           |            |     |
| 7                    | 0.006 | 192.168.10.103 | 192.168.10.20  | TCP       | 52811 → 80 | 54  |
| [FIN, ACK]           |       |                |                |           |            |     |
| 8                    | 0.006 | 192.168.10.20  | 192.168.10.103 | TCP       | 80 → 52811 | 54  |
| [ACK]                |       |                |                |           |            |     |

Détail de la trame 4 :

```

Trame 4 : 183 octets
Ethernet II
    Destination : 02:4d:44:4a:00:20
    Source : 02:4d:44:4a:01:03
    Type : IPv4 (0x0800)
Internet Protocol Version 4
    Source : 192.168.10.103
    Destination : 192.168.10.20
    Protocole : TCP (6)
Transmission Control Protocol
    Port source : 52811
    Port destination : 80
    Drapeaux : PSH, ACK
Données de l'application : 129 octets (nom du protocole masqué)
0036  47 45 54 20 2f 65 74 61 74 2e 68 74 6d 6c 20 48  GET /etat.html H
0046  54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 31  TTP/1.1..Host: 1
0056  39 32 2e 31 36 38 2e 31 30 2e 32 30 0d 0a 55 73  92.168.10.20..Us
0066  65 72 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c  er-Agent: Mozill
...

```

## Capture 2

Relevée à 14 h 40

| N°                | Temps | Source               | Destination    | Transport | Ports       | Lg |
|-------------------|-------|----------------------|----------------|-----------|-------------|----|
| Début des données |       |                      |                |           |             |    |
| -----             |       |                      |                |           |             |    |
| 1                 | 0.000 | 192.168.10.101       | 198.51.100.44  | TCP       | 52914 → 443 | 66 |
| [SYN]             |       |                      |                |           |             |    |
| 2                 | 0.018 | 198.51.100.44        | 192.168.10.101 | TCP       | 443 → 52914 | 66 |
| [SYN, ACK]        |       |                      |                |           |             |    |
| 3                 | 0.018 | 192.168.10.101       | 198.51.100.44  | TCP       | 52914 → 443 | 54 |
| [ACK]             |       |                      |                |           |             |    |
| 4                 | 0.019 | 192.168.10.101       | 198.51.100.44  | TCP       | 52914 → 443 |    |
| 287               |       | .....foi....f        |                |           |             |    |
| 5                 | 0.040 | 198.51.100.44        | 192.168.10.101 | TCP       | 443 → 52914 |    |
| 1413              |       | ....z...v...}.0f.... |                |           |             |    |
| 6                 | 0.041 | 192.168.10.101       | 198.51.100.44  | TCP       | 52914 → 443 |    |
| 118               |       | .....5.h.On....      |                |           |             |    |
| 7                 | 0.042 | 192.168.10.101       | 198.51.100.44  | TCP       | 52914 → 443 |    |
| 471               |       | .....[@.....\$.3     |                |           |             |    |
| 8                 | 0.061 | 198.51.100.44        | 192.168.10.101 | TCP       | 443 → 52914 | 54 |
| [ACK]             |       |                      |                |           |             |    |
| 9                 | 0.101 | 198.51.100.44        | 192.168.10.101 | TCP       | 443 → 52914 |    |
| 1309              |       | .....<_.....k...}    |                |           |             |    |
| 10                | 0.101 | 192.168.10.101       | 198.51.100.44  | TCP       | 52914 → 443 | 54 |
| [ACK]             |       |                      |                |           |             |    |
| 11                | 0.102 | 198.51.100.44        | 192.168.10.101 | TCP       | 443 → 52914 |    |
| 949               |       | ....zU&. .i.N...B..# |                |           |             |    |
| 12                | 0.102 | 192.168.10.101       | 198.51.100.44  | TCP       | 52914 → 443 | 54 |
| [ACK]             |       |                      |                |           |             |    |

Détail de la trame 4 :

```

Trame 4 : 287 octets
Ethernet II
  Destination : 02:4d:44:4a:00:01
  Source : 02:4d:44:4a:01:01
  Type : IPv4 (0x0800)
Internet Protocol Version 4
  Source : 192.168.10.101
  Destination : 198.51.100.44
  Protocole : TCP (6)
Transmission Control Protocol
  Port source : 52914
  Port destination : 443
  Drapeaux : PSH, ACK
Données de l'application : 233 octets (nom du protocole masqué)
0036  16 03 01 00 e4 01 00 00 e0 03 03 8a 66 6f 69 cb  ....foi.
0046  d6 08 a3 66 2a ba 6a 60 47 d4 87 5d 12 e3 d0 b6  ...f*.j`G..]....
0056  88 b3 67 56 9b 63 eb 3a dc bd 20 20 52 68 85 bf  ..gV.c:... Rh..
0066  2d 90 0c 7c 8a c6 d4 0b 13 21 29 ce cc 14 20 ca  -..|.....!)... .
...

```

## Capture 3

Relevée à 14 h 45

| N°                | Temps | Source         | Destination    | Transport | Ports      | Lg  |
|-------------------|-------|----------------|----------------|-----------|------------|-----|
| Début des données |       |                |                |           |            |     |
| --                | ----- | -----          | -----          | -----     | -----      | --- |
| 1                 | 0.000 | 192.168.10.105 | 192.168.10.1   | UDP       | 61022 → 53 |     |
| 87                | :     | .....www.fed   |                |           |            |     |
| 2                 | 0.012 | 192.168.10.1   | 192.168.10.105 | UDP       | 53 → 61022 |     |
| 103               | :     | .....www.fed   |                |           |            |     |

Détail de la trame 1 :

```

Trame 1 : 87 octets
Ethernet II
    Destination : 02:4d:44:4a:00:01
    Source : 02:4d:44:4a:01:05
    Type : IPv4 (0x0800)
Internet Protocol Version 4
    Source : 192.168.10.105
    Destination : 192.168.10.1
    Protocole : UDP (17)
User Datagram Protocol
    Port source : 61022
    Port destination : 53
Données de l'application : 45 octets (nom du protocole masqué)
002a  3a 7c 01 00 00 01 00 00 00 00 00 00 03 77 77 77  :|.....www
003a  0f 66 65 64 65 72 61 74 69 6f 6e 2d 6a 65 75 78  .federation-jeux
004a  07 65 78 61 6d 70 6c 65 00 00 01 00 01          .example.....

```

## Capture 4

Relevée à 14 h 51. La connexion était déjà ouverte : la capture commence au milieu de l'échange.

| N°                | Temps | Source                | Destination    | Transport | Ports       | Lg |
|-------------------|-------|-----------------------|----------------|-----------|-------------|----|
| Début des données |       |                       |                |           |             |    |
| 1                 | 0.000 | 192.168.10.105        | 192.168.10.10  | TCP       | 53120 → 445 |    |
| 280               |       | .....SMB..v..K!..k.   |                |           |             |    |
| 2                 | 0.001 | 192.168.10.10         | 192.168.10.105 | TCP       | 445 → 53120 |    |
| 230               |       | .....SMB6}...}.z{...  |                |           |             |    |
| 3                 | 0.003 | 192.168.10.105        | 192.168.10.10  | TCP       | 53120 → 445 |    |
| 306               |       | .....SMB.NL.....-.)L; |                |           |             |    |
| 4                 | 0.004 | 192.168.10.10         | 192.168.10.105 | TCP       | 445 → 53120 |    |
| 362               |       | ...0.SMB....[...p..p  |                |           |             |    |
| 5                 | 0.006 | 192.168.10.105        | 192.168.10.10  | TCP       | 53120 → 445 |    |
| 208               |       | .....SMBh.%......D.   |                |           |             |    |
| 6                 | 0.007 | 192.168.10.10         | 192.168.10.105 | TCP       | 445 → 53120 |    |
| 1320              |       | .....SMB.-.n.2FF..\.  |                |           |             |    |
| 7                 | 0.007 | 192.168.10.105        | 192.168.10.10  | TCP       | 53120 → 445 | 54 |
| [ACK]             |       |                       |                |           |             |    |

Détail de la trame 1 :

```

Trame 1 : 280 octets
Ethernet II
    Destination : 02:4d:44:4a:00:10
    Source : 02:4d:44:4a:01:05
    Type : IPv4 (0x0800)
Internet Protocol Version 4
    Source : 192.168.10.105
    Destination : 192.168.10.10
    Protocole : TCP (6)
Transmission Control Protocol
    Port source : 53120
    Port destination : 445
    Drapeaux : PSH, ACK
Données de l'application : 226 octets (nom du protocole masqué)
0036  00 00 00 de fd 53 4d 42 92 0c 76 d4 b9 4b 03 21  .....SMB..v..K.!
0046  03 bd 6b 01 4a 7e 54 64 88 66 91 d1 f8 8b 04 e0  ..k.J~Td.f.....
0056  a5 48 88 00 00 00 00 00 aa 00 00 00 00 00 01 00  .H.....
0066  45 00 00 00 00 30 00 00 f1 dc 86 74 aa fd 2d d2  E....0.....t...-
...

```

## Capture 5

Relevée à 15 h 03

| N°    | Temps       | Source               | Destination    | Transport | Ports      | Lg  |
|-------|-------------|----------------------|----------------|-----------|------------|-----|
| Début | des données |                      |                |           |            |     |
| ---   | -----       | -----                | -----          | -----     | -----      | --- |
| 1     | 0.000       | 192.168.10.102       | 192.168.10.10  | TCP       | 53201 → 22 | 66  |
|       |             | [SYN]                |                |           |            |     |
| 2     | 0.001       | 192.168.10.10        | 192.168.10.102 | TCP       | 22 → 53201 | 66  |
|       |             | [SYN, ACK]           |                |           |            |     |
| 3     | 0.001       | 192.168.10.102       | 192.168.10.10  | TCP       | 53201 → 22 | 54  |
|       |             | [ACK]                |                |           |            |     |
| 4     | 0.004       | 192.168.10.10        | 192.168.10.102 | TCP       | 22 → 53201 | 94  |
|       |             | SSH-2.0-OpenSSH_9.6p |                |           |            |     |
| 5     | 0.004       | 192.168.10.102       | 192.168.10.10  | TCP       | 53201 → 22 | 87  |
|       |             | SSH-2.0-OpenSSH_for_ |                |           |            |     |
| 6     | 0.005       | 192.168.10.102       | 192.168.10.10  | TCP       | 53201 → 22 |     |
| 358   |             | ...,...1.+g....&s... |                |           |            |     |
| 7     | 0.005       | 192.168.10.10        | 192.168.10.102 | TCP       | 22 → 53201 |     |
| 358   |             | ...,...m.}...&.n...  |                |           |            |     |
| 8     | 0.006       | 192.168.10.102       | 192.168.10.10  | TCP       | 53201 → 22 |     |
| 102   |             | ...,.....Km.E.a..z   |                |           |            |     |
| 9     | 0.007       | 192.168.10.10        | 192.168.10.102 | TCP       | 22 → 53201 |     |
| 262   |             | .....3....ssh-ed     |                |           |            |     |
| 10    | 0.008       | 192.168.10.102       | 192.168.10.10  | TCP       | 53201 → 22 |     |
| 70    |             | .....                |                |           |            |     |
| 11    | 0.008       | 192.168.10.102       | 192.168.10.10  | TCP       | 53201 → 22 | 98  |
|       |             | t....E[q.;i... ...   |                |           |            | ./  |
| 12    | 0.009       | 192.168.10.10        | 192.168.10.102 | TCP       | 22 → 53201 | 98  |
|       |             | z..c.....gu.         |                |           |            |     |
| 13    | 0.010       | 192.168.10.102       | 192.168.10.10  | TCP       | 53201 → 22 |     |
| 122   |             | ....1.3V,V..b.U..K.% |                |           |            |     |
| 14    | 0.011       | 192.168.10.10        | 192.168.10.102 | TCP       | 22 → 53201 |     |
| 114   |             | ..E..T.".....:1qP.y  |                |           |            |     |
| 15    | 1.831       | 192.168.10.102       | 192.168.10.10  | TCP       | 53201 → 22 |     |
| 186   |             | ..o.W.v{.ie...}..8.. |                |           |            |     |
| 16    | 1.833       | 192.168.10.10        | 192.168.10.102 | TCP       | 22 → 53201 |     |
| 106   |             | .F/. /I...T.....bz=. |                |           |            |     |
| 17    | 4.243       | 192.168.10.102       | 192.168.10.10  | TCP       | 53201 → 22 | 90  |
|       |             | ...&5.....~}.....z.  |                |           |            |     |
| 18    | 4.253       | 192.168.10.10        | 192.168.10.102 | TCP       | 22 → 53201 |     |
| 762   |             | ..W....U.-t.p0g>.n.0 |                |           |            |     |
| 19    | 4.253       | 192.168.10.102       | 192.168.10.10  | TCP       | 53201 → 22 | 54  |
|       |             | [ACK]                |                |           |            |     |

Détail de la trame 4 :

```

Trame 4 : 94 octets
Ethernet II
    Destination : 02:4d:44:4a:01:02
    Source : 02:4d:44:4a:00:10
    Type : IPv4 (0x0800)
Internet Protocol Version 4
    Source : 192.168.10.10
    Destination : 192.168.10.102
    Protocole : TCP (6)
Transmission Control Protocol
    Port source : 22
    Port destination : 53201
    Drapeaux : PSH, ACK
Données de l'application : 40 octets (nom du protocole masqué)
0036  53 53 48 2d 32 2e 30 2d 4f 70 65 6e 53 53 48 5f  SSH-2.0-OpenSSH_
0046  39 2e 36 70 31 20 55 62 75 6e 74 75 2d 33 75 62  9.6p1 Ubuntu-3ub
0056  75 6e 74 75 31 33 0d 0a                          untu13..

```

Capture 6

Relevée à 16 h 02

| N°                | Temps      | Source       | Destination     | Transport | Ports   | Lg |
|-------------------|------------|--------------|-----------------|-----------|---------|----|
| Début des données |            |              |                 |           |         |    |
| -----             |            |              |                 |           |         |    |
| 1                 | 0.000      | 0.0.0.0      | 255.255.255.255 | UDP       | 68 → 67 |    |
| 342               | ....^..... |              |                 |           |         |    |
| 2                 | 0.002      | 192.168.10.1 | 255.255.255.255 | UDP       | 67 → 68 |    |
| 342               | ....^..... |              |                 |           |         |    |
| 3                 | 0.003      | 0.0.0.0      | 255.255.255.255 | UDP       | 68 → 67 |    |
| 342               | ....^..... |              |                 |           |         |    |
| 4                 | 0.004      | 192.168.10.1 | 255.255.255.255 | UDP       | 67 → 68 |    |
| 342               | ....^..... |              |                 |           |         |    |

Détail de la trame 1 :

```

Trame 1 : 342 octets
Ethernet II
  Destination : ff:ff:ff:ff:ff:ff
  Source : f6:ce:23:0b:71:9a
  Type : IPv4 (0x0800)
Internet Protocol Version 4
  Source : 0.0.0.0
  Destination : 255.255.255.255
  Protocole : UDP (17)
User Datagram Protocol
  Port source : 68
  Port destination : 67
Données de l'application : 300 octets (nom du protocole masqué)
002a  01 01 06 00 5e 1a 93 c4 00 00 00 00 00 00 00 00  ....^.....
003a  00 00 00 00 00 00 00 00 00 00 00 00 00 f6 ce 23 0b  .....#.
...
011a  35 01 01 3d 07 01 f6 ce 23 0b 71 9a 0c 0a 4c 41  5..=....#.q...LA
012a  50 54 4f 50 2d 4c 45 41 37 0e 01 03 06 0f 1f 21  PTOP-LEA7.....!
013a  2b 2c 2e 2f 77 79 f9 fc ff 00 00 00 00 00 00 00  +,./wy.....
...

```

## A5 — La table des ports usuels

*Source : registre des numéros de ports de l'IANA, l'organisme qui attribue les numéros sur Internet, traduit et raccourci. Plage des ports des clients : documentation de Microsoft.*

*Colonne Wireshark : relevée sur Wireshark 4.6. Utilisée dans : LABO étapes 13 à 16.*

| Port   | Transport        | Protocole | Service rendu                                              | Ce que Wireshark affiche |
|--------|------------------|-----------|------------------------------------------------------------|--------------------------|
| 20, 21 | TCP              | FTP       | transfert de fichiers, en clair                            | FTP                      |
| 22     | TCP              | SSH       | piloter un serveur à distance, chiffré                     | SSH, SSHv2               |
| 23     | TCP              | Telnet    | piloter un serveur à distance, en clair                    | TELNET                   |
| 25     | TCP              | SMTP      | envoyer un courriel                                        | SMTP                     |
| 53     | UDP, parfois TCP | DNS       | résolution de noms : trouver l'adresse IP d'un nom de site | DNS                      |

| Port   | Transport | Protocole | Service rendu                                                                                                             | Ce que Wireshark affiche |
|--------|-----------|-----------|---------------------------------------------------------------------------------------------------------------------------|--------------------------|
| 67, 68 | UDP       | DHCP      | donner une adresse IP à une machine qui arrive sur le réseau. Le serveur écoute sur le port 67, le client sur le port 68. | DHCP                     |
| 80     | TCP       | HTTP      | pages web, en clair                                                                                                       | HTTP                     |
| 123    | UDP       | NTP       | mettre les horloges à l'heure                                                                                             | NTP                      |
| 143    | TCP       | IMAP      | lire ses courriels sur le serveur                                                                                         | IMAP                     |
| 161    | UDP       | SNMP      | surveiller les équipements réseau                                                                                         | SNMP                     |
| 443    | TCP       | HTTPS     | pages web, protégées par TLS                                                                                              | TLSv1.2, TLSv1.3         |
| 445    | TCP       | SMB       | partage de fichiers et d'imprimantes Windows                                                                              | SMB, SMB2                |
| 631    | TCP       | IPP       | impression                                                                                                                | IPP                      |
| 3306   | TCP       | MySQL     | base de données                                                                                                           | MySQL                    |
| 3389   | TCP       | RDP       | bureau à distance Windows                                                                                                 | RDP                      |

### Trois remarques

- Les ports 0 à 1023 sont réservés aux services connus. Le serveur les garde toujours.
- Un client Windows choisit son port au hasard, entre 49152 et 65535. Ce port change à chaque conversation.
- Un service peut être réglé sur un autre port. Le port est un indice, pas une preuve.

## A6 — Les machines de la maquette MaisonDesJeux

Source : document de scénario, § 5 « Plan d'adressage », « Nommage » et « Captures de référence ». Utilisée dans : LABO étapes 21 et 22.

Réseau 192.168.10.0/24. Les adresses MAC du parc suivent une règle : 02:4d:44:4a , puis 00 pour l'infrastructure ou 01 pour un poste, puis le numéro de la machine.

| Nom                                 | Rôle                                                                                                | Adresse IP                      | Adresse MAC                                                         |
|-------------------------------------|-----------------------------------------------------------------------------------------------------|---------------------------------|---------------------------------------------------------------------|
| <b>RTR-MDJ-01</b>                   | routeur d'accès à Internet ; il rend aussi les services DNS et DHCP au réseau local                 | 192.168.10.1                    | 02:4d:44:4a:00:01                                                   |
| <b>SRV-MDJ-01</b>                   | serveur de fichiers de l'association                                                                | 192.168.10.10                   | 02:4d:44:4a:00:10                                                   |
| <b>IMP-MDJ-01</b>                   | imprimante réseau                                                                                   | 192.168.10.20                   | 02:4d:44:4a:00:20                                                   |
| <b>MDJ-PC-01</b> à <b>MDJ-PC-12</b> | postes fixes : <b>MDJ-PC-nn</b> a l'adresse .1nn ; par exemple, 192.168.10.105 est <b>MDJ-PC-05</b> | 192.168.10.101 à 192.168.10.112 | 02:4d:44:4a:01:01 à 02:4d:44:4a:01:12                               |
| portables des bénévoles             | adresse donnée automatiquement                                                                      | 192.168.10.150 à 192.168.10.199 | leur propre adresse MAC ; celle de Léa Fournier : f6:ce:23:0b:71:9a |

## Les adresses sur Internet

| Nom du site                        | Adresse IP    |
|------------------------------------|---------------|
| <b>www.federation-jeux.example</b> | 203.0.113.80  |
| <b>www.banque-mutuelle.example</b> | 198.51.100.44 |

## Les adresses spéciales

| Adresse           | Ce qu'elle veut dire                                         |
|-------------------|--------------------------------------------------------------|
| 0.0.0.0           | « je n'ai pas encore d'adresse »                             |
| 255.255.255.255   | « tout le réseau local »                                     |
| ff:ff:ff:ff:ff:ff | l'adresse MAC qui touche toutes les machines du réseau local |