TD1 - Prise en main de Wireshark

Pascal Cotret, pascal.cotret@ensta-bretagne.fr

2 septembre 2022

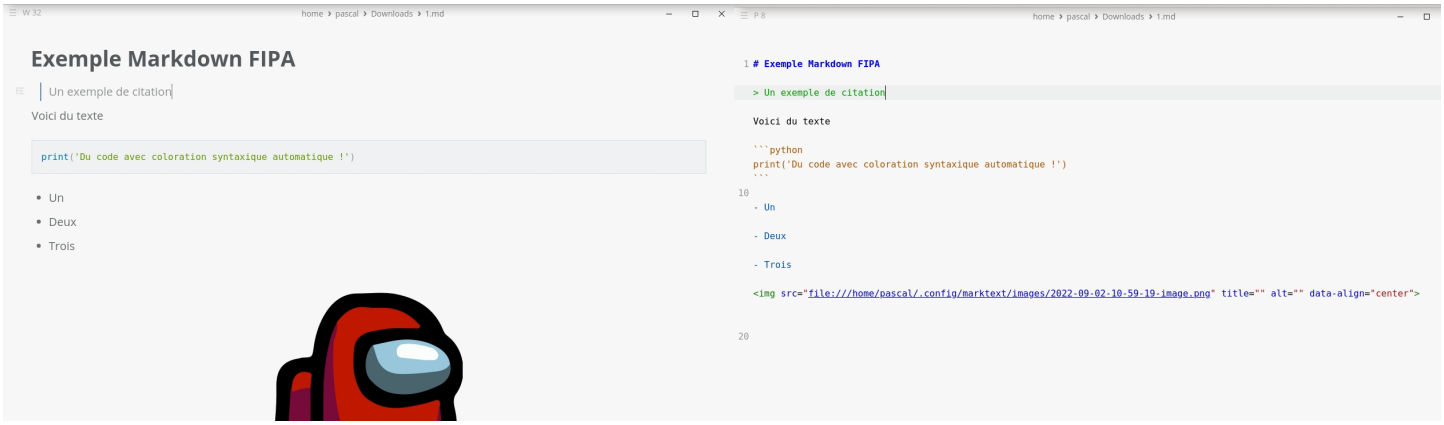
Table des matières

Remarques

Pour ce TD (et plus globalement les TD/TP de ce cours d'introduction sur les réseaux), il est plus que recommandé de prendre des notes. Sans faire un compte-rendu en bonne et due forme, c'est le meilleur moyen d'assimiler les notions vues en cours et de pouvoir réviser un peu plus tard pour préparer l'examen.

Je suis un grand amateur de Markdown, vous le verrez en temps voulu. C'est le langage utilisé sur Gitlab par exemple. Quelques ressources :

— Un tutoriel sur la syntaxe https://www.markdowntutorial.com/
— Marktext, un éditeur WYSIWYG open-source : https://github.com/marktext/marktext/releases/tag/v0.17.1. Ça exporte même en HTML et en PDF !



1 Présentation de Wireshark

Wireshark[1] est un outil de capture et d'analyse de paquets réseau libre publié sous licence GPL. Créé par Gerald Combs en 1999 sous le nom originel d'Ethereal, cet outil est développé en C/C++ par de nombreux contributeurs[2]. Il utilise la bibliothèque d'interface graphique QT[3] et la bibliothèque de capture de paquet libpcap[4]. Cette dernière est développée par les auteurs de tcpdump, un autre logiciel libre de

1. https://www.wireshark.org/
2. https://www.wireshark.org/about.html#authors
3. https://www.qt.io/
4. http://www.tcpdump.org/

capture de paquets, fonctionnant pour sa part uniquement en ligne de commande.

Wireshark est disponible pour différents systèmes d'exploitation : Windows, GNU/Linux, les systèmes BSD et Mac OS X. Le projet est aujourd'hui majoritairement sponsorisé par Riverbed[5], l'employeur de Gerald Combs. Cette société est spécialisée dans les réseaux informatique et notamment dans l'analyse des réseaux et de leurs performances.

Wireshark propose différentes fonctionnalités :
- Il permet de capturer le trafic en utilisant une interface réseau Ethernet, une interface Wifi ou un port USB et d'enregistrer ce trafic sous forme de fichier au format pcap ;
- Il permet de visualiser et d'analyser du trafic préalablement capturé et enregistré dans un fichier ;
- Il permet de modifier les paquets enregistrés au préalable dans un fichier pcap.

Le site web du projet fournit une documentation conséquente[6], notamment un guide de l'utilisateur[7], des pages de manuel pour les différents outils disponibles en ligne de commande[8], un wiki[9] et des vidéos présentant certains aspects de Wireshark.

2 Installation de Wireshark

Il est possible de télécharger différentes versions de Wireshark sur la page de téléchargement du site web du projet[10]. Le site fournit directement des programmes d'installation pour Windows et MacOSX. L'application est également disponible dans la plupart des systèmes de gestion de paquets utilisés par les distributions GNU/Linux, les systèmes BSD et MacOSX (via Homebrew, MacPorts ou Fink). Les plus courageux peuvent s'aventurer à télécharger le code source du logiciel, disponible sur le serveur de version du projet[11], et à le compiler.

Nous vous recommandons d'utiliser les moyens d'installation suivants :
- Pour Windows, utiliser l'installateur automatique qui intègre l'installation de la bibliothèque Winpcap ;
- Pour MacOSX, utiliser l'archive dmg ;
- Pour Linux, utiliser le système d'installation de paquets de votre distribution.

L'utilisation de la fonction de capture de paquets nécessite des droits particuliers que ne possède généralement par l'utilisateur de l'ordinateur[12]. Suivant les systèmes, il peut être nécessaire d'octroyer ces droits à l'utilisateur pour qu'il puisse capturer des paquets.
- Sous Windows, si vous acceptez les options par défaut de l'installateur, celui-ci installe automatique la bibliothèque WinPcap (cf Figure 1) et propose d'exécuter automatiquement le pilote de cette bibliothèque au démarrage de la machine (cf Figure 2) avec les droits nécessaires ;
- Sous Mac OSX, l'installation via l'archive DMG positionne les droits automatiquement pour l'utilisateur courant ;
- Sous Linux, la plupart des distributions vont créer un groupe spécifique possédant les droits de capture (par exemple, sous Debian, il s'agit du groupe `wireshark`). Il suffit alors d'ajouter l'utilisateur à ce groupe à l'aide de la commande :

5. https://www.riverbed.com/fr/
6. https://www.wireshark.org/docs/
7. https://www.wireshark.org/docs/wsug_html/
8. https://www.wireshark.org/docs/man-pages/
9. https://wiki.wireshark.org/
10. https://www.wireshark.org/#download
11. https://code.wireshark.org/review/#/admin/projects/wireshark
12. https://wiki.wireshark.org/CaptureSetup/CapturePrivileges

`sudo usermod -a -G wireshark user` (en remplaçant `user` par le login utilisateur) ;

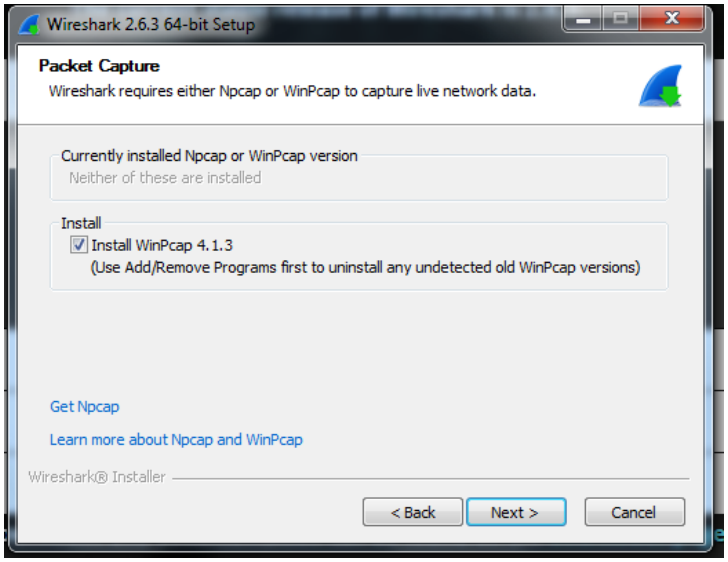


FIGURE 1 – Installation de la bibliothèque WinPcap

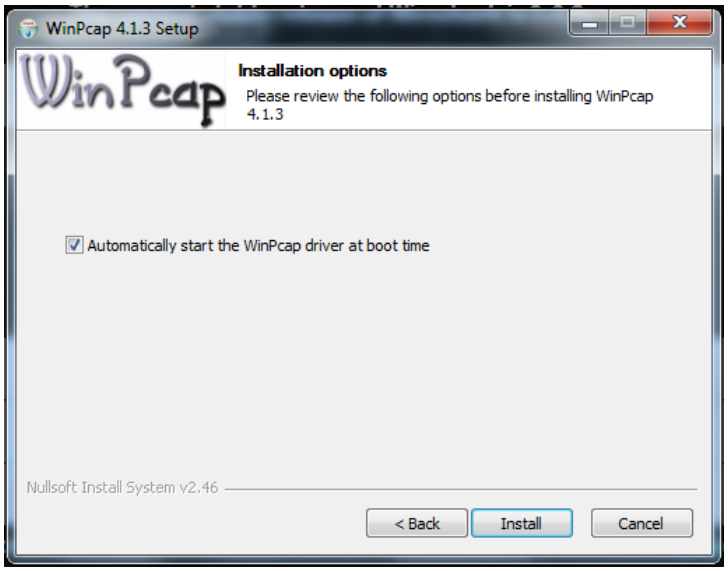


FIGURE 2 – Exécution automatique du pilote de la bibliothèque WinPcap

En résumé, les procédures d'installation par défaut doivent normalement vous permettre de réaliser des captures. Si ce n'est pas le cas, il vous faudra suivre la procédure décrite sur la page : https://wiki.wireshark.org/CaptureSetup/CapturePrivileges.

3 Première exécution

Lorsque vous exécutez Wireshark, vous devriez obtenir la fenêtre principale illustrée par la figure 3. Cette fenêtre vous indique les différentes interfaces de captures disponibles. Vous devriez retrouver notamment vos différentes interfaces réseau (Ethernet, Wifi, etc.).

L'interface de *loopback* est une interface "virtuelle" présentée par le système d'exploitation. Les applications qui s'exécutent sur une même machine peuvent l'utiliser pour échanger de l'information en utilisant les protocoles réseau courants (IP, TCP, UDP) sans pour autant avoir besoin d'utiliser une carte réseau réelle. Évidemment elle ne permet pas de communiquer avec d'autres machines.

Selon la configuration de votre machine, d'autres interfaces peuvent apparaître (par exemple, les interfaces correspondant aux réseaux virtuels de Virtualbox).

Wireshark affiche un graphique à côté de chaque interface. Ce graphique est mis à jour en temps réel et indique l'activité (c'est-à-dire la quantité de paquets échangés) de chaque interface.

Plusieurs icônes de la barre d'outils permettent d'accéder rapidement à des fonctionnalités qui sont également proposées dans la barre de menu, notamment :

— Démarrer une capture sur l'interface actuellement sélectionnée.
— Stopper une capture en cours.
— Choisir et configurer une interface de capture.
— Ouvrir un fichier de capture préalablement sauvegardé.
— Sauvegarder la capture, etc.

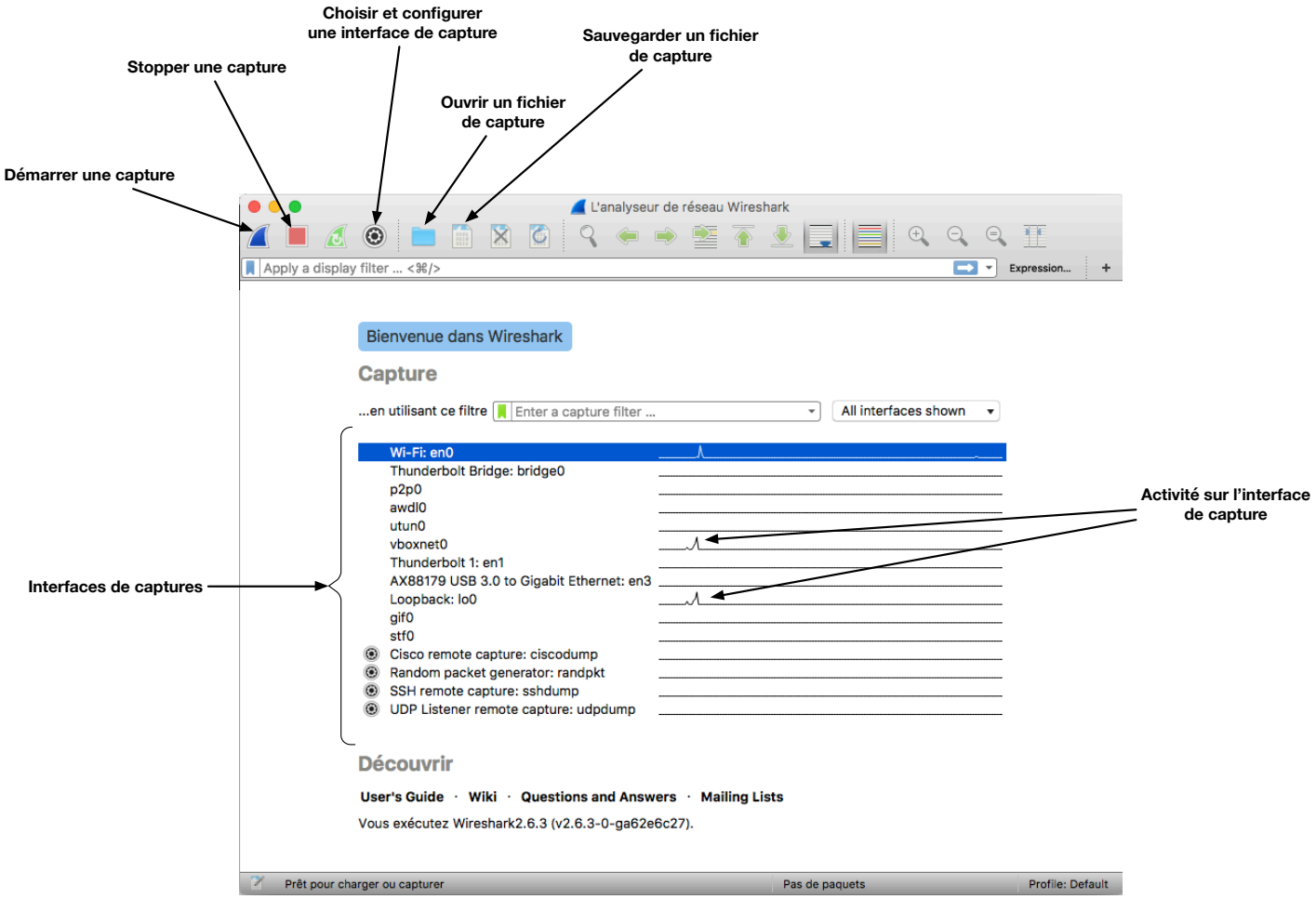


FIGURE 3 – Fenêtre principale de Wireshark

3.1 Configuration de l'interface de capture

Lorsque vous voulez réaliser une capture, nous vous recommandons de choisir explicitement et de configurer l'interface de capture au préalable. Vous devriez alors obtenir la fenêtre illustrée par la figure 4. Cette fenêtre liste les interfaces de capture disponibles. Il est alors possible de sélectionner l'interface qui sera utilisée pour la capture.

Il est également possible de configurer les interfaces. Vous pouvez notamment activer le mode *promiscuous* qui placera l'interface dans le mode correspondant. Dans ce mode particulier, l'interface réseau capture l'ensemble des paquets qui parviennent jusqu'à elle, y compris les paquets qui ne lui sont pas destinés. La plupart des cartes Ethernet accepte sans problème de se placer dans ce mode.

En revanche, cela est plus compliqué pour les interfaces Wifi. Pour capturer l'ensemble des paquets émis par des stations à portée radio, il est nécessaire de placer la carte en mode *monitor*, ce qui n'est pas supporté par toutes les cartes Wifi. En outre, le support peut dépendre du type de système d'exploitation. Le lecteur intéressé pourra consulter la page de documentation relative à ce sujet [13]. Par défaut, si le mode moniteur n'est pas activé, l'interface Wifi ne capture que les paquets à destination de la machine ou émis par celle-ci. En outre, les informations sur la couche de liaison Wifi 802.11 ne sont pas disponibles. Wireshark ne dispose que des adresses MAC source et destination qu'il affiche dans une pseudo couche de liaison Ethernet.

De même, la capture des paquets utilisés par le protocole USB (par exemple entre votre clavier et l'ordinateur), nécessite des pilotes spécifiques qui ne sont pas toujours installés par défaut (ce n'est par exemple pas le cas sous Windows). Le lecteur intéressé pourra consulter la page de documentation à relative à ce sujet [14].

Wireshark supporte deux types de filtre :

1. les filtres de capture ;
2. les filtres d'affichage.

Nous utiliserons essentiellement les seconds. Cependant, quelques explications à propose de chacun d'eux s'imposent.

Tout d'abord, les filtres de capture sont hérités de la librairie libpcap du projet tcpdump. Utiliser un tel filtre permet de programmer de manière très efficace le pilote de périphérique de la carte réseau afin de ne capturer et donc enregistrer que les paquets vérifiant certaines conditions précises et réduit donc grandement la quantité de données à analyser ultérieurement dans Wireshark. Cependant, lorsque l'on ne sait pas exactement ce que l'on cherche dans le trafic, ou lorsque l'on cherche à comprendre l'origine d'un problème réseau sans avoir d'idée précise de son origine, il est préférable de capturer le trafic dans sa globalité (le fonctionnement par défaut de Wireshark). Les filtres de capture utilisent une syntaxe bien particulière qui est peut être consultée sur le site du projet tcpdump [15].

Les filtres d'affichage permettent quant à eux (et comme leur nom l'indique) de réduire la quantité de paquets affichés à un moment donné, en ne se concentrant que sur ceux vérifiant une propriété donnée. Là encore, ces filtre s'expriment à l'aide de règle, dont la syntaxe est malheureusement différentes de ceux de capture. Nous donnons quelques détails à propos de ces filtres à la section 3.2.

Question 1. Sélectionnez une interface de capture Ethernet ou Wifi et démarrez une capture. Ouvrez votre navigateur et spécifiez l'adresses (URL) suivante :
http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html. Validez de manière à af-

13. https://wiki.wireshark.org/CaptureSetup/WLAN
14. https://wiki.wireshark.org/CaptureSetup/USB
15. http://www.tcpdump.org/manpages/pcap-filter.7.html

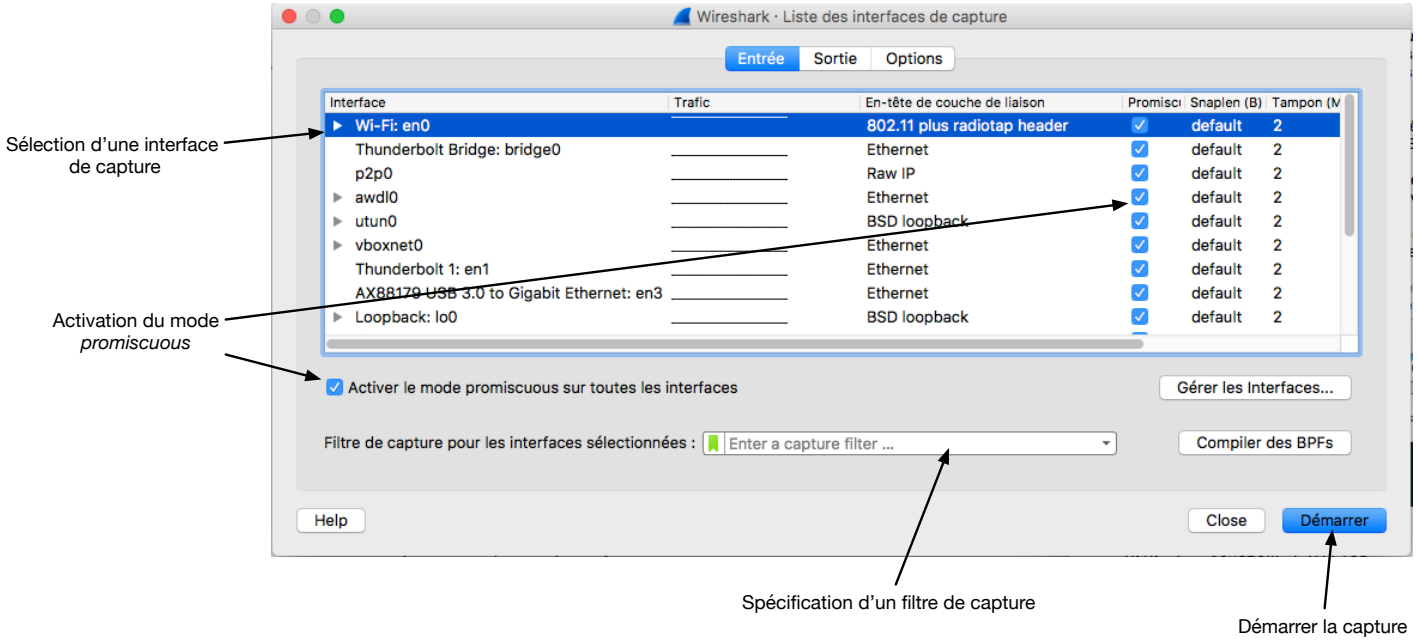


FIGURE 4 – Fenêtre de configuration des interfaces

ficher la page Web correspondante.

3.2 Analyse des paquets capturés

Lorsque vous lancez une capture, vous devriez obtenir une fenêtre similaire à celle illustrée par la figure 5. La zone située au milieu de la fenêtre affiche en temps réel la liste des paquets capturés. Par défaut, la fenêtre défile automatiquement sur les derniers paquets capturés (ce comportement est modifiable dans la configuration de la capture). Chaque paquet est décrit par son numéro dans la capture, les adresses source et destination, le protocole utilisé ainsi qu'un cours descriptif du paquet. Les adresses correspondent aux adresses de plus haut niveau que Wireshark a pu identifier. Il s'agit donc des adresses IP si le paquet contient une en-tête IP ou à défaut des adresses MAC. Pour ces dernières, Wireshark identifie automatiquement le constructeur de la carte réseau via le préfixe de l'adresse MAC.

La zone en bas détaille le contenu du paquet sélectionné. Lors de la capture, il s'agit du dernier paquet capturé. Il est possible d'afficher le contenu de l'en-tête de chacun des protocoles utilisés dans le paquet (principe d'encapsulation). Une fois la capture arrêtée, il est possible de sélectionner n'importe quel paquet et d'inspecter son contenu.

Différents outils permettent de naviguer parmi les paquets capturés comme l'illustre la figure 6 :

— Les flèches en vert permettent de sélectionner le paquet suivant ou précédent, le premier ou le dernier paquet de la capture ansi que de sélectionner un paquet selon son numéro ;

— La loupe permet de rechercher un paquet selon sa description sommaire dans la liste d'affichage ou son contenu. Il est possible de rechercher des chaînes de caractères selon différents encodages (ASCII, UTF-8, etc.), des valeurs d'octet exprimées en hexadécimal ou des motifs de recherche sous forme d'expressions régulières [16] ;

— Il est également possible de spécifier un filtre d'affichage qui permet de sélectionner les paquets qui vérifient un critère de filtre. Ce dernier est exprimé sous la forme d'une formule logique sur la valeur des champs des différents protocoles.

16. https://fr.wikipedia.org/wiki/Expression_r%C3%A9guli%C3%A8re

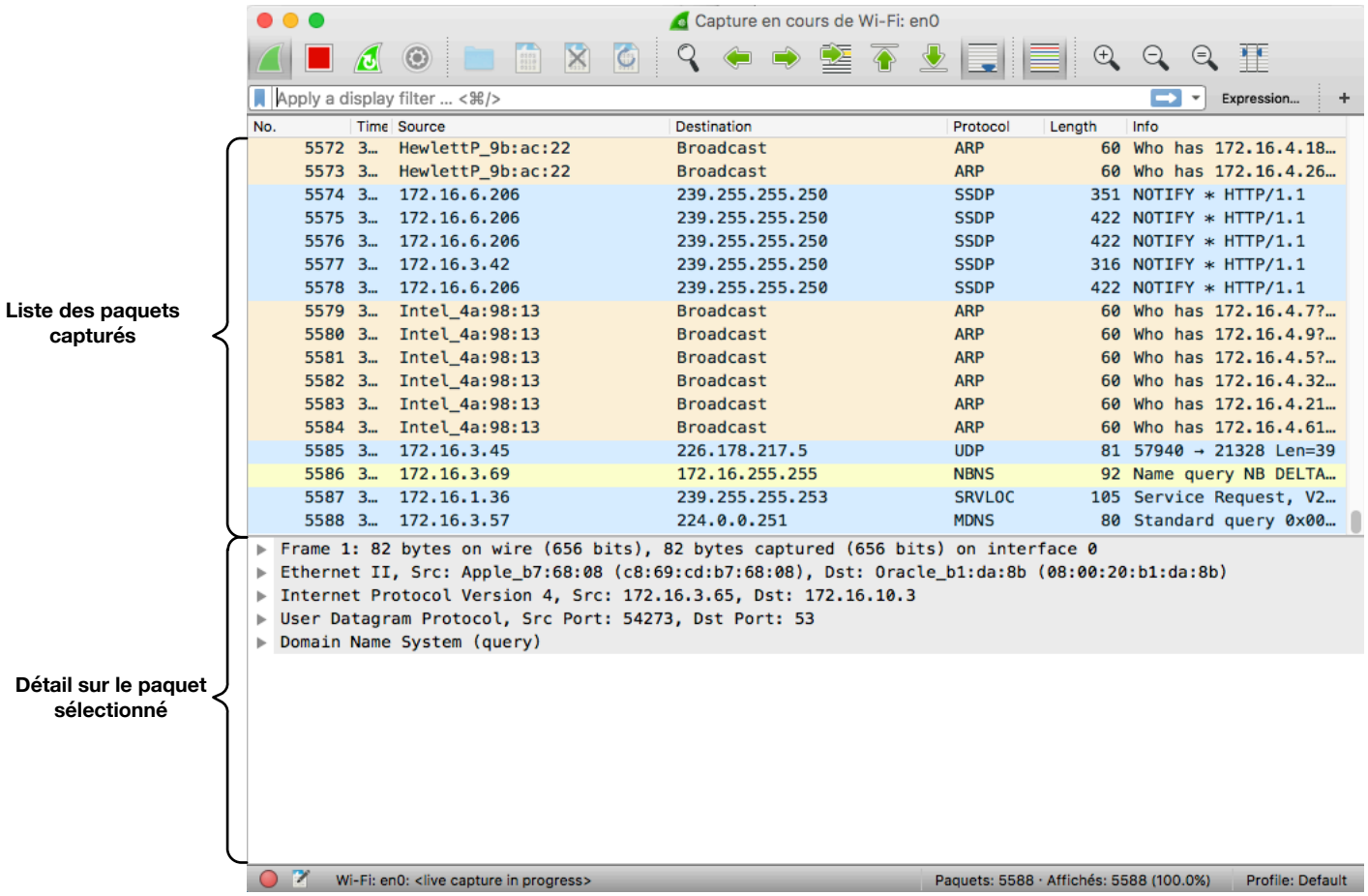


FIGURE 5 – Capture des paquets

Lorsqu'un paquet est sélectionné, Wireshark affiche parfois un lien avec d'autres paquets capturés correspond à la requête (si le paquet sélectionné est une réponse) ou la réponse (si le paquet sélectionné est une réponse). Ces liens sont matérialisés par des flèches situées à gauche de la zone centrale.

Les filtres d'affichage sont un outil puissant et souvent plus précis que la fonction de recherche. Il faut néanmoins connaître la syntaxe de ces filtres. Pour vous aider à rédiger votre filtre, Wireshark propose un éditeur qui liste notamment les différents noms des champs qui peuvent être filtrés. Pour afficher cet éditeur, il faut cliquer sur `Expression...` à droite de la barre d'affichage des filtres. Vous obtiendrez alors une fenêtre similaire à celle illustrée par la figure 7.

Vous trouverez à gauche de cette fenêtre la liste de l'ensemble des champs à partir desquels il est possible de filtrer, regroupés par protocole. Pour afficher la liste des champs d'un protocole, il suffit de cliquer sur le triangle situé à gauche de chaque protocole. À droite, vous trouverez l'ensemble des relations qu'il est possible d'utiliser sur le champs sélectionné ainsi que la valeur du filtre. Par exemple sur la figure 7 nous avons filtré le champ `ip.addr` du protocole IPv4 en appliquant une relation d'égalité (`==`) à la valeur `192.168.10.4`. Il est également possible de filtrer des paquets selon les protocole utilisés. Ainsi le filtre `http` permet de n'afficher que les paquets qui utilisent le protocole HTTP.

À tout moment, le champ décrivant le filtre, tout comme la barre de filtre de la fenêtre principale, surligne le filtre en vert s'il respecte la syntaxe et en rouge s'il ne la respecte pas. Le coloriage est actualisé en temps réel. Ne soyez donc pas surpris si le champ apparaît en rouge lorsque vous commencez de spécifier votre filtre. Il le restera tant que vous n'aurez pas spécifier complètement votre filtre en respectant la syntaxe.

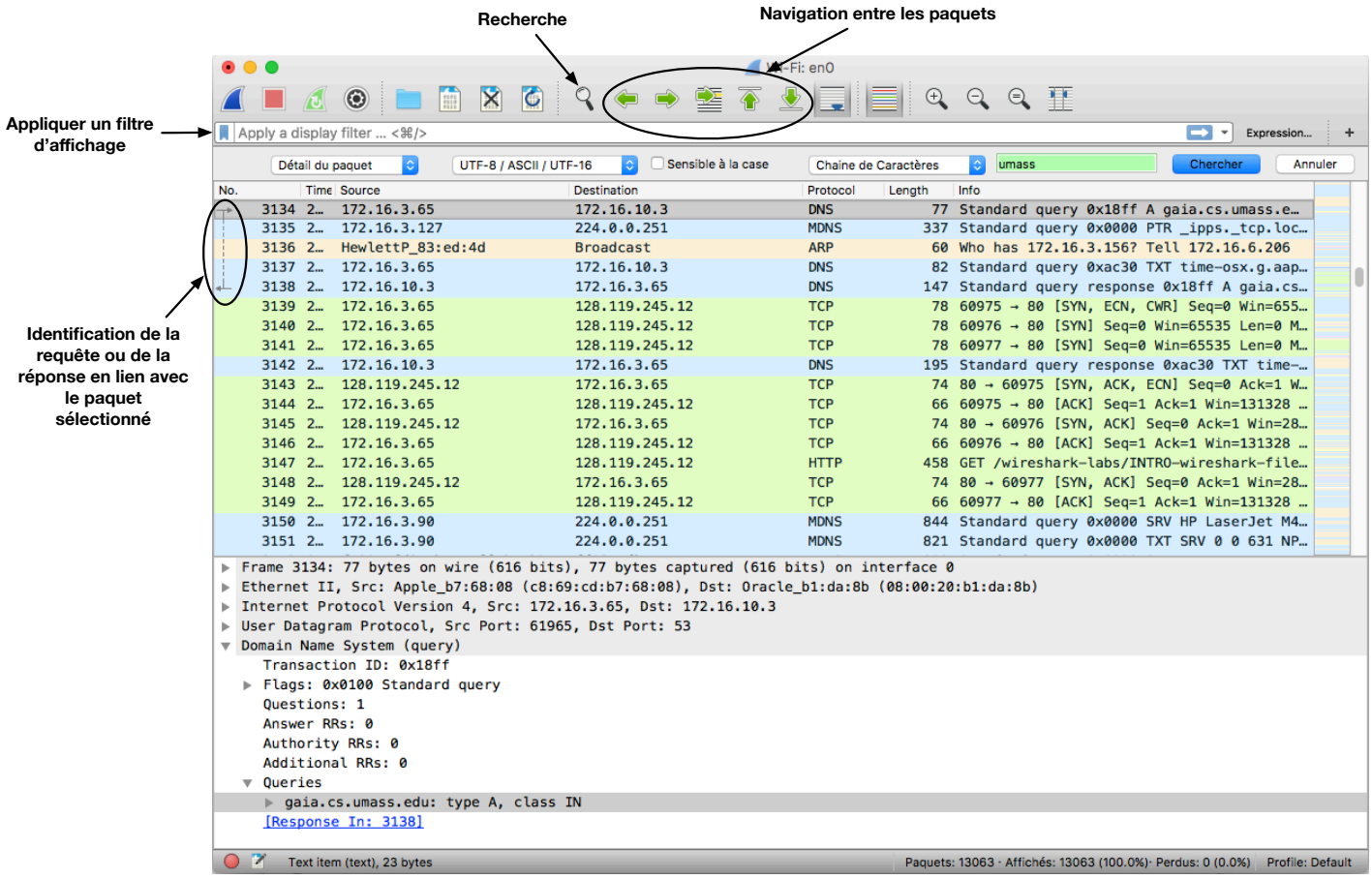


FIGURE 6 – Capture des paquets

Il est possible de combiner plusieurs filtres selon des conjonctions (ET logique, noté `&&`) ou des disjonctions (OU logique, noté `||`). Il est également possible d'inverser un filtre à l'aide de l'opérateur de négation (`!`) situé avant une expression. Ainsi le filtre `!ip.src == 192.168.10.4` affiche tous les paquets sauf ceux émis par l'adresse IP 192.168.10.4. L'utilisation de parenthèses permet de définir précisément l'ordre d'évaluation. Ansi, le filtre suivant filtre tous les paquets émis depuis l'adresse IP 128.119.245.12 ou à destination de cette adresse IP et utilisant le protocole HTTP :

```
(ip.dst == 128.119.245.12 || ip.src == 128.119.245.12) && http
```

Attention, lorsqu'une expression est validée, elle est simplement ajoutée à la barre de filtre mais le filtre n'est pas appliqué. Pour appliquer le filtre, il faut cliquer sur la flèche située à droite de la barre des filtres.

Question 2. Stoppez la capture et, à l'aide des filtres d'affichage, filtrer les paquets à destination de l'adresse IP 128.119.245.12 et qui utilisent le protocole HTTP.

4 Quelques questions pour s'entraîner

Question 3. Dans la trace capturée, identifiez l'adresse IP et l'adresse MAC de votre machine. Vérifiez

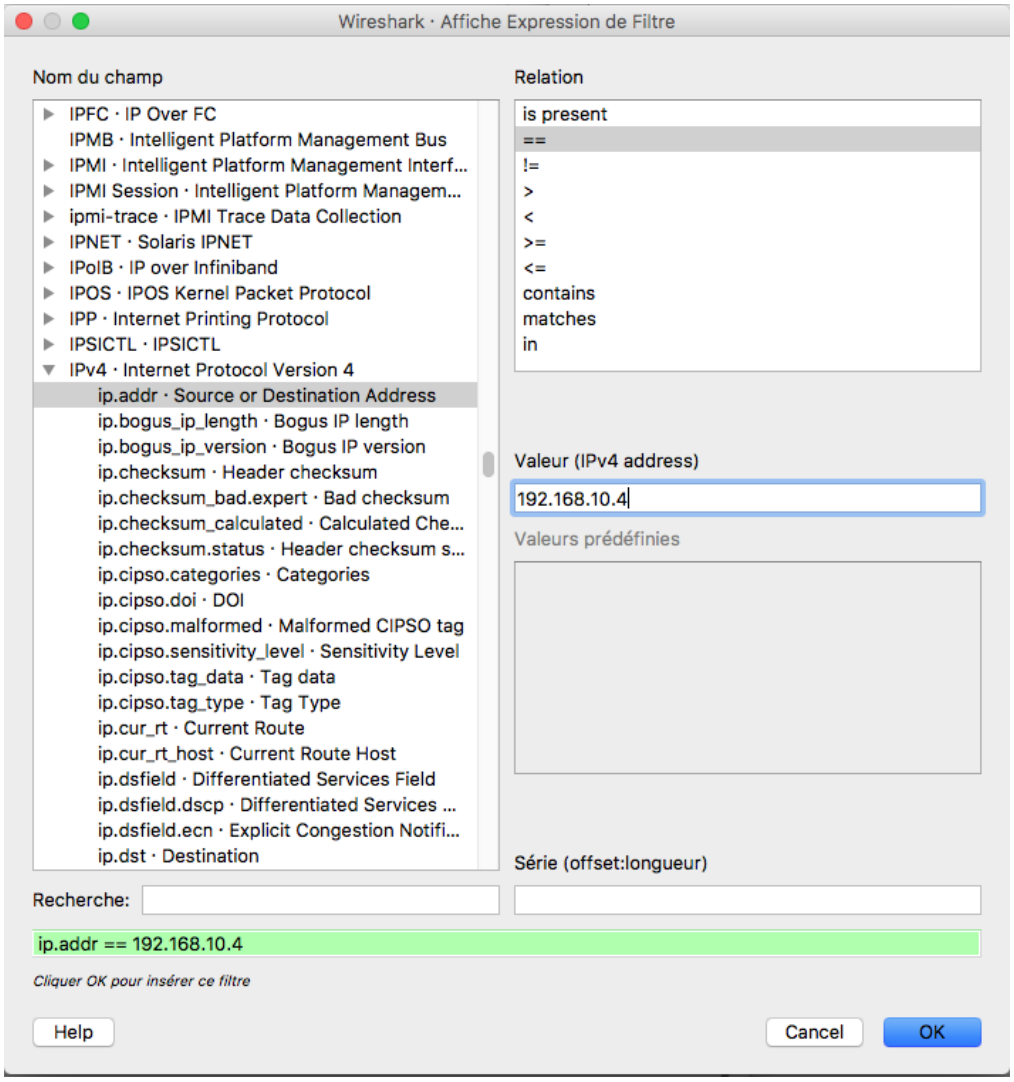


FIGURE 7 – Capture des paquets

que ces valeurs sont cohérentes avec la configuration réseau de votre machine. Sous Linux et MacOSX, vous pouvez obtenir la configuration réseau à l'aide de la commande `ifconfig` exécutée dans un terminal. Sous Windows, vous pouvez l'obtenir par la commande `ipconfig` exécutée dans un terminal (programme `cmd.exe`)

Question 4. Quelle est l'adresse IP du serveur `gaia.cs.umass.edu` ?

Question 5. Identifiez la requête HTTP GET envoyée par votre navigateur Web au serveur `gaia.cs.umass.edu`. Quels sont les différents protocoles utilisés dans le paquet transportant cette requête ?

Question 6. Combien de temps s'est écoulé entre la requête de votre navigateur et la réponse HTTP sur serveur (OK) ?

Références

[1] *The RISC-V Instruction Set Manual - Volume I : Unprivileged ISA*. https://github.com/riscv/riscv-isa-manual/releases/download/Ratified-IMAFDQC/riscv-spec-20191213.pdf.