2 TP 02 – ANALYSE DU TRAFIC RESEAU (1/2).

Une des meilleures façons pour comprendre le fonctionnement d'un protocole dans le réseau est de l'observer en action, et de voir les messages échangés par deux entités protocolaires. Pour cela, on lance l'exécution de certaines actions (un scénario) par le protocole et on observer les conséquences. Ceci peut se faire via simulation ou par une exécution réelle. Dans ce TP, nous allons observer quelques protocoles réseau "en action", en train d'interagir et d'échanger des messages dans une exécution réelle, et ceci en utilisant des logiciels appelés *Renifleur* ou *Sniffer*.

2.1 PREREQUIS

Des notions sur les protocoles réseaux et leurs rôles en général est nécessaire pour ce TP.

2.2 LOGICIELS RENIFLEURS (SNIFFER)

Un logiciel *renifleur* ou *sniffer* est un logiciel d'analyse de paquets. Il peut intercepter, enregistrer et analyser les données transitant au sein d'un réseau local *non-commuté*. Ainsi, il permet de capturer les données du *flux* traversant le réseau, et afficher le contenu de chacun des champs constituant ces données conformément aux spécifications de chacun des protocoles utilisés. Noter qu'une donnée est structurée en un ensemble de champs, chacun contient une information bien précise (voir Annexe).

Un sniffer peut être utilisé même pour résoudre les problèmes réseaux possibles (sécurité, etc.) en visualisant et analysant ce qui passe à travers les interfaces réseau. Le logiciel sniffer que nous allons utiliser dans nos TP est le *Wireshark*.

2.3 POURQUOI WIRESHARK ?

Les caractéristiques de Wireshark qui nous ont encouragé à l'adopter pour nos TP sont les suivants :

1. Grâce à ses fonctionnalités, il a la capacité d'afficher et d'analyser le contenu des messages envoyés (ou reçu) par des centaines de protocoles de toutes les couches de la pile protocolaire.
2. Il fonctionne sous différents systèmes d'exploitation (Windows, Linux, Mac, etc.). Aussi, il est capable de lire les données sur différentes interfaces réseaux (Ethernet, LAN sans fil 802.11, etc.).
3. Il est stable, et il a une interface utilisateur bien conçue et conviviale.
4. Il est gratuit et bien documenté (des guides utilisateur, des FAQ détaillée, etc.).
5. Il a une large communauté d'utilisateurs.

2.4 OBJECTIF DE CE TP

Dans ce premier TP, sur Wireshark, l'étudiant va se familiariser avec ce logiciel. Ainsi il va apprendre à capturer les paquets de données transitant dans le réseau, les observer et les analyser.

2.5 WIRESHARK

Une fois installé sur un ordinateur, Wireshark peut être utilisé pour intercepter les messages envoyés de (ou reçus par) par cet ordinateur, afficher leur contenu tout en précisant les champs de chacun des protocoles utilisés dans cet échange. Noter que Wireshark n'envoie aucun message, il ne fait qu'observer les messages échangés par les applications et les protocoles en cours d'exécution sur un ordinateur.

2.5.1 Principe de fonctionnement

Le principe du fonctionnement de Wireshark est illustré dans la Figure 4. Il est constitué de deux parties : *(i)* un module qui capture une copie de chaque message, envoyé ou reçu par l'ordinateur, en utilisant la bibliothèque de capture de paquets (pcap), *(ii)* un module qui fait l'analyse des données capturées et précise le contenu de tous les champs d'un message.

Il capture des trames de la couche liaison (Ethernet) d'un ordinateur, ce qui permettra de capturer tous les messages envoyés (ou reçus) par tous les protocoles exécutés sur cet ordinateur. Ceci car la communication se fait selon le principe d'*encapsulation*, et donc tous les protocoles des couches supérieures sont finalement encapsulés dans une trame Ethernet. Grâce à l'analyseur de paquets, qui comprend la structure des messages échangés par tous les protocoles, Wireshark peut afficher le contenu de chaque champ d'un message dépendant du protocole qu'il l'a échangé.

Exemple. Supposons qu'on a utilisé le navigateur pour consulter le site web du centre universitaire de Mila. Donc, au niveau « Application » on a utilisé le protocole HTTP. Le message HTTP est encapsulé dans des messages TCP ou UDP, qui sont à leurs tour encapsulés dans des paquets IP, encapsulés par la suite dans des trames Ethernet, et enfin transmis sur le support physique.

Wireshark capture une trame Ethernet. Comme il sait bien le format de cette trame, il peut identifier le datagramme IP encapsulé dedans. Aussi, il connait également le format du datagramme IP, et ainsi il peut extraire le segment TCP. Enfin, il connait la structure du segment TCP, et donc il peut extraire le message HTTP qu'il contient. Enfin, il analyse le message HTTP et l'affiche selon la structure de la donnée HTTP.

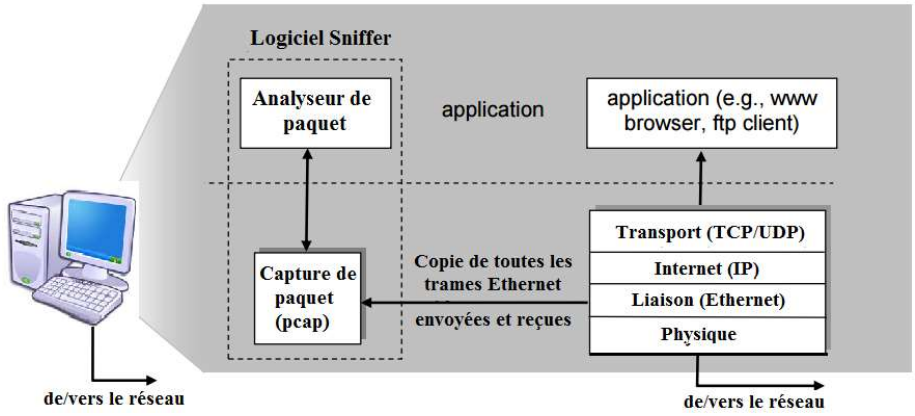


Figure 4. *Architecture du Wireshark.*

2.5.2 Téléchargement et installation

La dernière version de Wireshark peut être téléchargée via le lien suivant : *http://www.wireshark.org/download.html*. La page principale de Wireshark: *http://www.wireshark.org* contient une FAQ et d'autres informations utiles qu'on peut y retourner en cas de problème d'installation ou d'exécution. Une fois téléchargé, l'installation peut être lancée en double-cliquant sur l'exécutable et en suivant les étapes. Pour une utilisation plus simple, il est conseillé de créer un raccourci de l'exécutable *Wireshark-gtk* dans le bureau.

N.B. Dans ce TP, nous utilisons le Wireshark 6 version 2.2.3.

2.5.3 Interface principale

Pour lancer Wireshark, cliquer sur *Wireshark-gtk-Raccourci* dans le bureau. Il s'ouvre sur l'interface suivante (voir Figure 5) composée de cinq zones :

(1) Le panneau de commande.

La partie haute de ce panneau contient les menus déroulants standards, les plus intéressants sont :

- ***Fichier*** *:* pour sauvegarder ou ouvrir un fichier contenant une capture de données.
- ***Capture*** *:* pour lancer et arrêter une capture de données.

En dessous des menus déroulant, il y a les icônes de lancement rapide des tâches présentées dans ces menus. Ensuite, il y a le champ ***Filtre***, pour filtrer les paquets capturés à afficher. On peut par exemple choisir de n'afficher que les paquets concernant le protocole IP en tapant dans cette zone le mot « IP ». Aussi il y a le bouton « Expression » juste à côté pour définir une expression du filtre plus complexe.

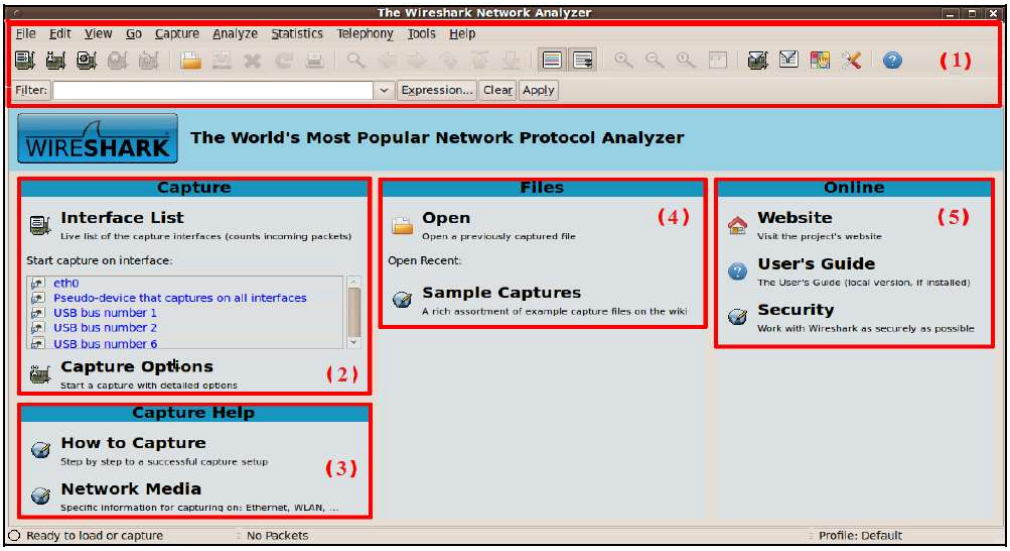


Figure 5. *Interface principale de Wireshark.*

(2) La liste des interfaces réseaux et lancement rapide d'une capture.

Ce panneau affiche toutes les cartes réseaux actives dans le PC y compris les cartes virtuelles (qui n'existent pas physiquement) installées par des logiciels comme Virtual Box.

(3) L'aide sur la capture de paquets.

Elle contient toute l'aide nécessaire pour apprendre à capturer et analyser les données.

(4) La liste des captures récentes (enregistrées).

Cette zone affiche la liste des fichiers contenant des captures réalisées récemment. Elle offre un accès rapide pour ouvrir une capture récente enregistrée.

(5) L'aide en ligne et le manuel de l'utilisateur.

Cette zone affiche des liens utiles comme celui de la page principale et celui du guide de l'utilisateur.

2.5.4 Capture des trames avec Wireshark

La capture se fait en trois étapes :

(1) Le choix de l'interface réseau sur laquelle Wireshark va capturer les données (noter qu'un ordinateur peut avoir plusieurs interfaces réseau).
(2) Le lancement de la capture.
(3) L'arrêt de la capture.

Pour ce faire, différentes manières existent, les plus pratiques sont les deux suivantes :

2.5.4.1 *En utilisant la zone « Capture »*

Dans la zone (2) "Liste des interfaces réseau" sont affichées les interfaces réseaux de l'ordinateur. Cliquer directement sur l'interface désirée. Ensuite, cliquer sur le bouton « Start » qui se trouve au-dessus.

2.5.4.2 *En utilisant le panneau de commande*

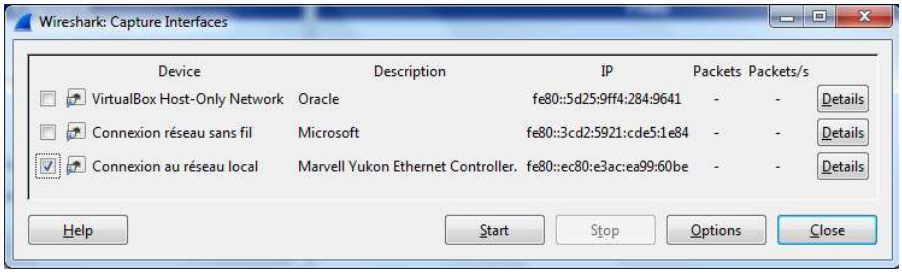
En utilisant la zone (1) « Panneau de commande », soit :

(i) En choisissant « Interfaces » dans le menu « Capture ».

(ii) En cliquant sur le premier bouton de la liste des icônes de lancement rapide, intitulé « Interface List »

(iii) En utilisant la combinaison clavier Ctrl + I.

La fenêtre suivante s'ouvre :



Il suffit de choisir l'interface désirée et puis cliquer sur le bouton « Start » pour lancer la capture.

Après le choix d'une interface et le lancement d'une capture, Wireshark commence la capture de tous les paquets envoyés à (ou reçu par) cette interface, et affiche une fenêtre comme celle-ci (Figure 6) :

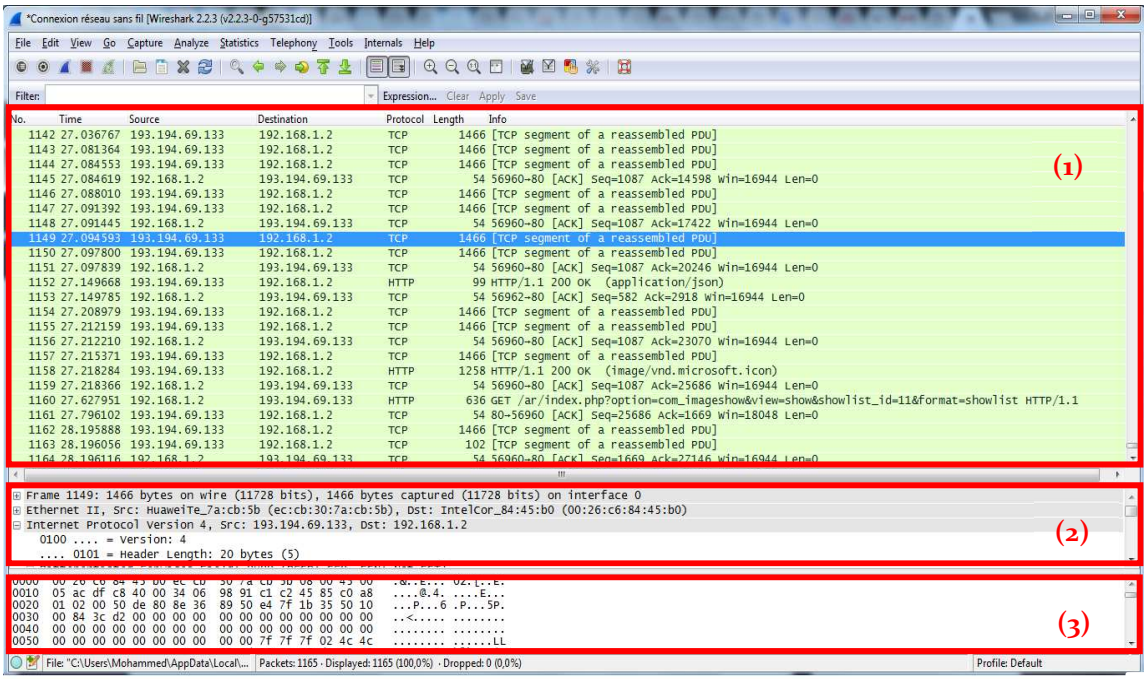


Figure 6. *Fenêtre de capture.*

L'interface est constituée de trois zones :

(1) *La zone (1) en haut :* affiche la liste des paquets capturés.

(2) *La zone (2) au milieu* : affiche le détail d'un paquet sélectionné (mis en évidence) dans la zone (1).

(3) *La zone (3) en bas :* affiche le contenu du paquet sélectionné dans la zone (1) en ASCII et en hexadécimal.

La capture peut être stoppée en cliquant tout simplement sur le bouton « Stop the running live capture » de la barre des icônes de lancement rapide dans le « Panneau de commande ».

2.5.5 Analyse des paquets capturés

Dans la fenêtre de capture, les informations affichées dans une zone inférieure correspondent à ce qui a été sélectionnée dans la zone supérieure. Ainsi, si on veut connaitre les détails sur un paquet capturé, il suffit de le sélectionné dans la zone (1), est ses informations d'encapsulation seront affichées dans la zone (2), ainsi que son contenu hexadécimal dans la zone (3).

2.5.5.1 Filtrer les paquets capturés

On peut remarquer qu'il y a beaucoup de types de paquets capturés et affichés par Wireshark, à cause du grand nombre de protocoles supportés par ce logiciel. Pour pouvoir chercher à l'intérieur des données capturées et n'afficher que le trafic qui nous intéresse, on utilise des filtres.

La syntaxe générale d'une expression simple d'un filtre est :

Protocole	•	Paramètre	Opérateur	Valeur

Exemple : ip . addr == 192.168.1.1

L'expression du filtre peut être définie visuellement en cliquant sur le bouton « Expression » à côté du champ « Filter » dans le panneau de commande, ce qui ouvre la fenêtre de la Figure 7.

Pour définir l'expression du filtre, suivre les étapes illustrées dans la figure 7 :

(1) Choisir le protocole, et pour le paramètre il suffit de dérouler la liste des paramètres en cliquant sur « > » devant le protocole et choisir un.
(2) Ensuite choisir l'opérateur de comparaison.
(3) Et enfin, mettre la valeur à considérer dans le champ valeur.

Tout au long des étapes, on voit construite l'expression tout en bas de la fenêtre. Une fois terminer, cliquer sur « OK ».

Noter que les étapes illustrées dans la Figure 7 correspondent à la définition du filtre de l'exemple *ip.addr == 192.168.1.1*.

On peut définir des expressions de filtre plus complexes en utilisant des opérateurs logiques. Le format est :

Expression	Opérateur logique	Expression	.....

Exemple : TCP || UDP

Quatre opérateurs logiques sont disponibles :

Opérateur	Définition
&&	ET logique
||	OU logique
^^	XOR logique
!	NON logique

Exemples de filtres :
ip.addr == 192.168.1.1 → Affiche les paquets avec une adresse source ou destination de 192.168.1.1
tcp || ip || dns → Affiche les trafics TCP ou IP ou DNS.
ip.src == 192.168.1.1 && ip.dst != 172.16.10.2 → Affiche les paquets avec une adresse IP source égale à 192.168.1.1 et de destination différente de 172.16.10.2.
eth.addr == 00 :2F : 4C :01 :23 :6C → affiche le trafic de la machine dont l'@ MAC est 00 :2F : 4C :01 :23 :6C.

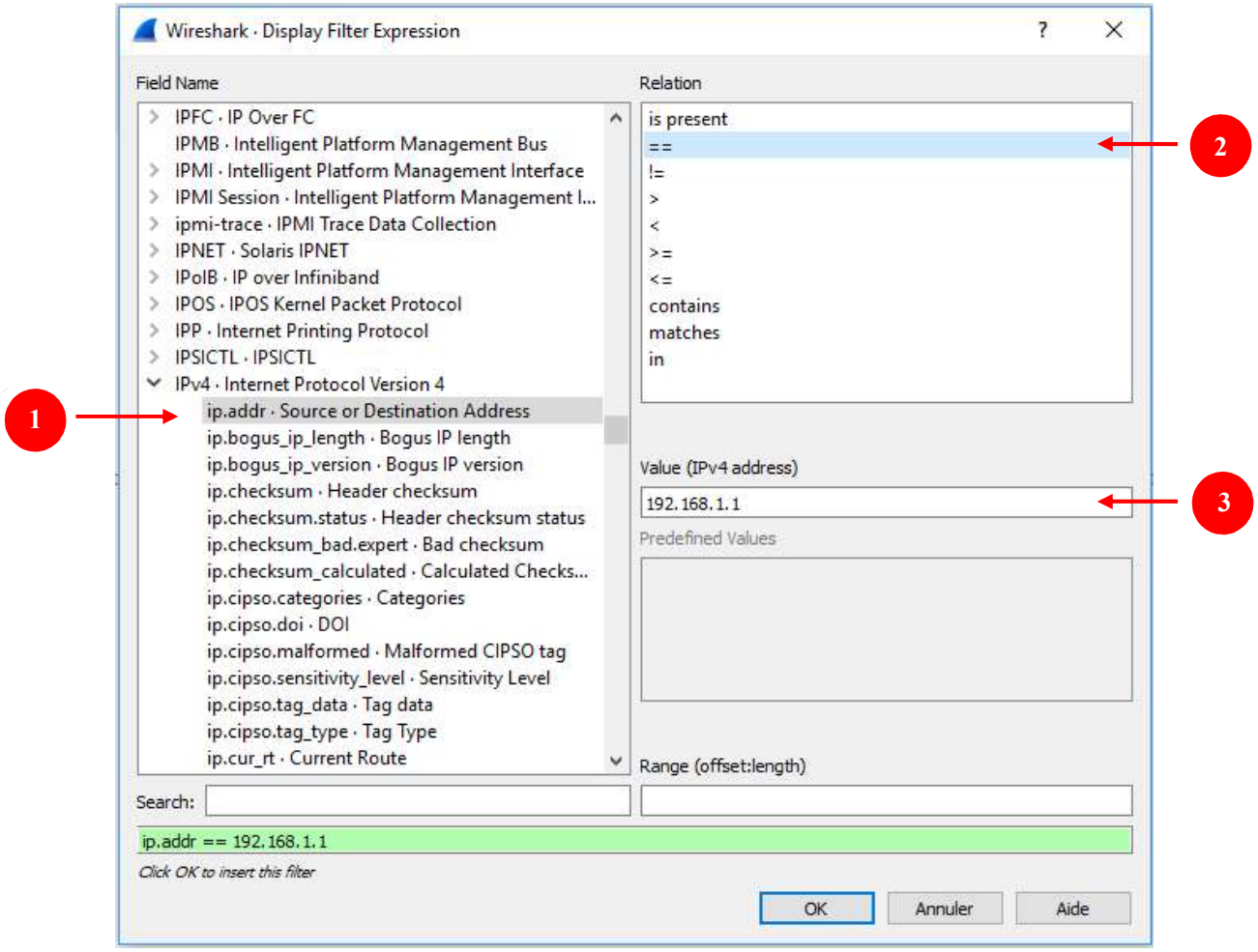


Figure 7. *Définition du filtre d'affichage.*

N.B. Ces filtres sont des filtres d'affichage, et il y a aussi les filtres de capture qui déterminent quels types de paquets à capturer.

2.5.5.2 *Le paquet capturé*

Dans la liste des paquets capturés affichés dans la zone (1), chaque ligne correspond à un paquet capturé. Elle contient le numéro de paquet attribué par Wireshark, l'heure à laquelle le paquet a été capturé, les adresses de source et de destination du paquet, le type de protocole, et des informations spécifiques au protocole contenues dans le paquet. La liste des paquets capturés peut être triée selon l'une de ces catégories en cliquant sur le nom de la colonne correspondante.

2.5.5.3 *Les détails sur le paquet sélectionné*

Si un paquet est sélectionné dans la zone (1), la zone (2) fournit les détails sur ce paquet ainsi que ses différents niveaux d'encapsulation.

Exemple : si on sélectionne un paquet de type HTTP dans la zone (1), la zone (2) affiche quelque chose similaire à ceci :

```
⊞ Frame 626: 613 bytes on wire (4904 bits), 613 bytes captured
⊞ Ethernet II, Src: IntelCor_84:45:b0 (00:26:c6:84:45:b0), Dst
⊞ Internet Protocol Version 4, Src: 192.168.1.2, Dst: 193.194.
⊞ Transmission Control Protocol, Src Port: 56971, Dst Port: 80
⊞ Hypertext Transfer Protocol
```

Chacune des entrées correspond à un niveau d'encapsulation. L'ordre d'encapsulation se lit de bas en haut dans la zone (3), voir Figure 7. Noter que dans la zone (3), c'est en quelque sorte l'ordre de dé-encapsulation qui est affiché. En effet, Wireshark capture une suite d'octets. Il extrait la trame de cette suite. Ensuite, il extrait le paquet IP de la trame, puis il extrait le message TCP (ou UDP) du paquet IP, et enfin il extrait le message http.

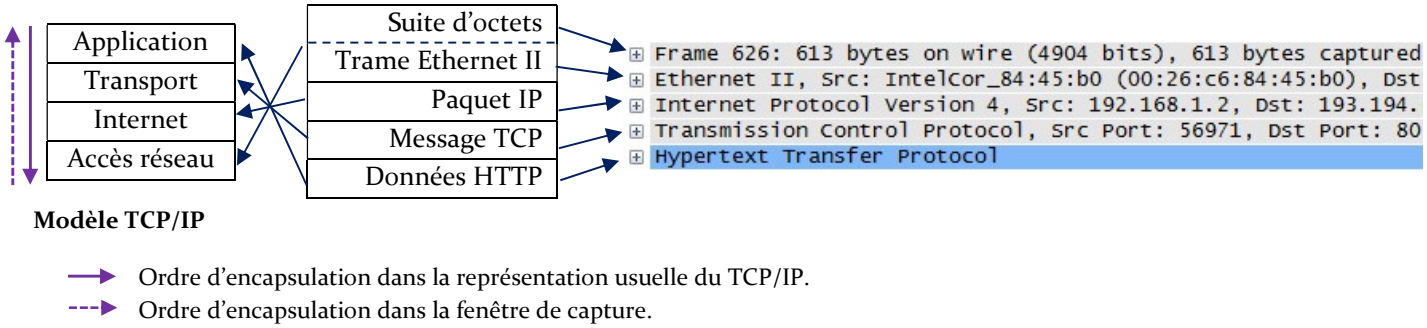
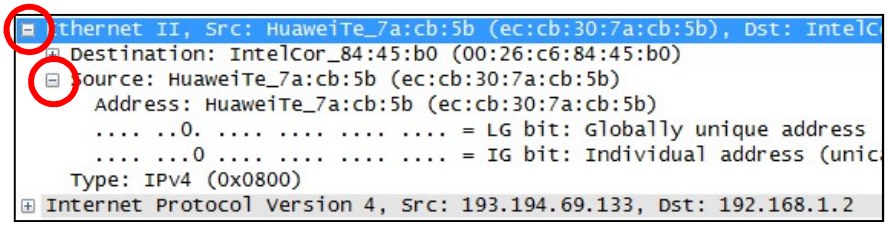


Figure 7. Ordre d'encapsulation dans la fenêtre de capture – zone (2).

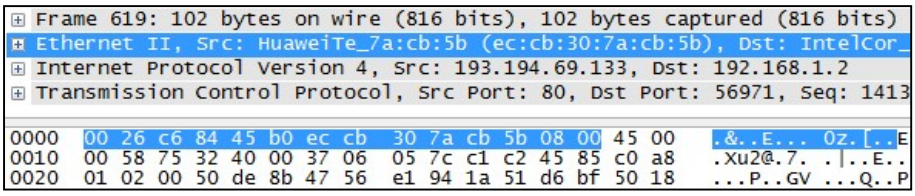
Noter que le clic sur le « + » devant chaque niveau d'encapsulation permet de visualiser l'ensemble des champs le composant. Aussi, certains champs sont à leur tour peuvent être déroulés.
Exemple : le déroulement de l'entrée Ethernet (couche liaison) permet de visualiser les champs « Destination », « Source » et « Type ». Les champs « Destination » et « Source » peuvent à leur tour être étendus.



2.5.5.4 *Le contenu du paquet sélectionné*

La zone (3) permet de visualiser le contenu de la trame capturée en hexadécimal. Un clic sur un niveau d'encapsulation de la zone (2) permet de visualiser la portion d'octets correspondante à ce niveau dans la zone (3). Inversement, un clic sur un octet quelconque de la zone (3) affiche le champ correspondant dans la zone (2).

Exemple : un clic sur le niveau Ethernet (couche liaison de données) mettra en évidence dans la zone (3) les octets correspondant. Ceci peut être fait pour chaque champ.



2.6 TRAVAIL DEMANDE

1. Lancez une capture à l'aide de Wireshark en sélectionnant l'interface « Ethernet ».
2. Énumérez 5 différents protocoles exécutés par votre PC.
3. Appliquer les filtres d'affichage suivants à votre capture :
   a) Seulement le trafic ARP.
   b) Le trafic TCP ou le trafic UDP entre les ports 30 et 60.
   c) Tous les messages TCP sauf ceux dont le port de source ou de destination est 80.
   d) Seulement les paquets envoyés par votre machine.
   e) Pas de trafic SSL.
4. Lancez une capture de paquets, et pendant que Wireshark est en cours d'exécution, lancez la commande « ping » vers une machine dans le réseau :
   a) Quels sont les paquets envoyés et reçus suite à l'exécution du « ping » ?
   b) Quel est le protocole utilisé ? A quelles couches appartient-il ?
5. Lancez une autre capture, et pendant que Wireshark est en cours d'exécution, entrez dans votre navigateur l'adresse du site du centre universitaire de Mila : *www.centre-univ-mila.dz*. Une fois que la page du site affichée par le navigateur, arrêtez la capture :
   a) Quels est le protocole utilisé par le navigateur pour communiquer avec le serveur du centre-univ-mila.dz ?
   b) Filtrer les paquets capturés et ne laissez affichés que les messages http.
   c) Sélectionnez un message HTTP dans la zone (1) :
      (i) Le protocole http est de quelle couche ?
      (ii) Quel protocole de transport utilisé par http ?
      (iii) Donnez les détail d'encapsulation du message HTTP.
   d) Quels sont les deux types de messages HTTP échangés entre votre PC et le serveur ?
   e) Combien de temps a-t-il fallu pour recevoir la réponse à la requête HTTP envoyée par votre PC ?
   f) Quelle-est l'adresse Internet du serveur du « centre-univ-mial.dz » ?