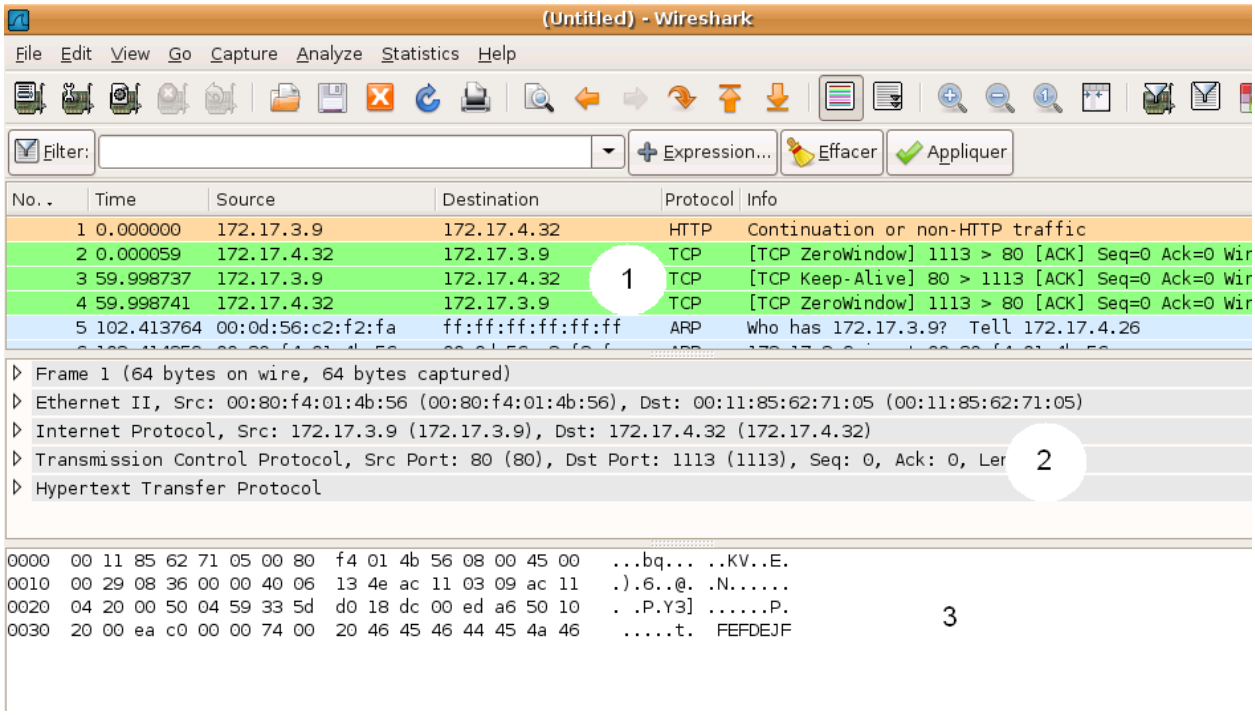
Annexe 2 – Wireshark

Sommaire :

I - Présentation

Wireshark est un outil sous licence **GNU** qui permet d'analyser finement le trafic réseau et d'interpréter la structure des paquets, cela de façon graphique. Il utilise la librairie **Libpcap** et la syntaxe des **filtres** est similaire à celle de la commande Unix **tcpdump**.

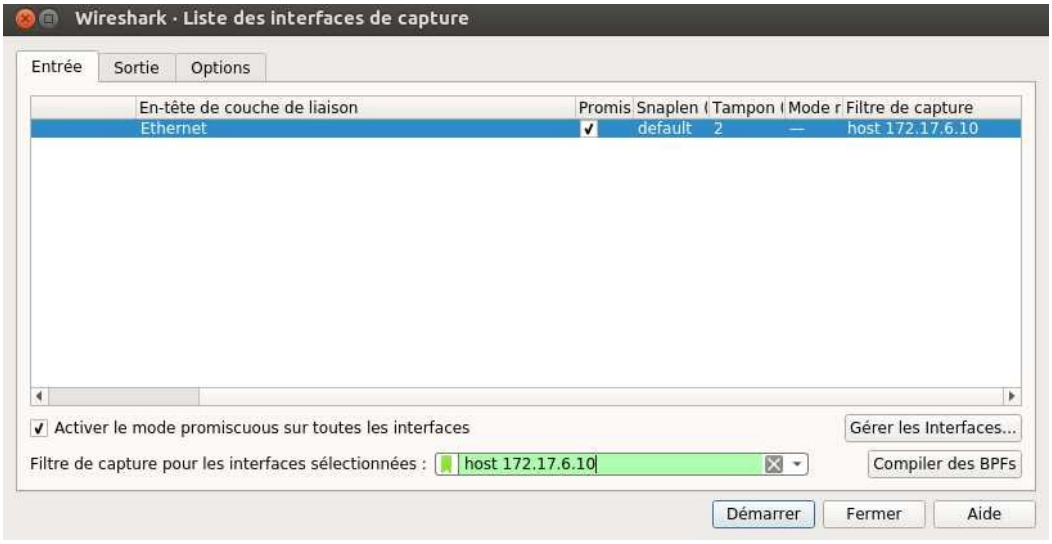
La fenêtre principale de **Wireshark** comporte trois volets :



- Le volet **1** permet de recenser **l'ensemble des paquets capturées**. Sont spécifiés **l'émetteur** de la trame, le **destinataire** de la trame et le **protocole réseau** mis en œuvre ;
- Le volet **2** permet de visualiser la **pile des protocoles** employés dans le **paquet** sélectionné dans le premier volet ;
- Le volet **3** permet de visualiser **l'ensemble du paquet capturé** au format **hexadécimal** et la traduction **ASCII** correspondante.

II - Capture de trames

Pour capturer des trames, il suffit de cliquer sur **Capture Options** (*menu Capture*). La fenêtre suivante apparaît :



Deux critères peuvent être spécifiés avant de lancer la capture en cliquant sur **Démarrer** :

- **Interface** : *permet de sélectionner l'interface physique (carte réseau, ...) à partir de laquelle la capture va être effectuée ;*
- **Filtre de capture** : *permet d'établir un filtre de capture (syntaxe tcpdump) ou d'appliquer un filtre sauvegardé.*

Il y a deux sortes de filtres : les filtres de **capture** et les filtres d'**affichage**. La syntaxe des filtres de capture est la même que les filtres utilisés pour la commande **tcpdump**.

Nous allons détailler ici seulement les filtres de **capture**. Ne seront gardés que les paquets pour lesquels le filtre est vrai. Les filtres se décomposent en 3 parties :

- le **protocole** qui peut être **arp**, **ether**, **fddi**, **icmp**, **ip**, **ip6**, **link**, **ppp**, **radio**, **rarp**, **slip**, **tcp**, **tr**, **udp** ou **wlan** ;
- la **direction** qui peut être **src** (source) ou **dst** (destination) ;
- un **champ** qui peut être **host**, **net** ou **port** suivi d'une valeur.

Les opérateurs **and** (ou **&&**), **or** (ou **||**) et **not** (ou **!**) peuvent être utilisés pour combiner des filtres. Voici quelques exemples de filtres de capture :

Filtre	Fonction
host 172.16.0.1	Ne conserve que les paquets à destination ou en provenance de la machine 172.16.0.1
host 172.16.0.1 and host 172.16.0.2	Ne conserve que les paquets à destination ou en provenance des machines 172.16.0.1 et 172.16.0.2
host 172.16.0.1 and tcp	Ne conserve que les paquets TCP à destination ou en provenance de la machine 172.16.0.1
udp port 53	Ne conserve que les paquets UDP en provenance ou en destination du port 53
udp port 53 and dst host 172.16.0.1	Ne conserve que les paquets UDP en provenance ou en destination du port 53 à destination de la machine 172.16.0.1
tcp dst port 80 and dst host 172.16.0.1 and src net 172.16.0.0 mask 255.255.255.0	Ne conserve que les paquets TCP en destination de la machine 172.16.0.1 sur le port 80 et en provenance des machines du réseau 172.16.0/24