

Capture, Filtrage et Analyse de trames ETHERNET avec le logiciel Wireshark - correction -

Wireshark est un programme informatique libre de droit, qui permet de capturer et d'analyser les trames d'information qui transitent par les interfaces de communication du terminal sur lequel il s'exécute. *Wireshark* est ainsi apparenté aux logiciels appelés « Sniffer » ou « analyseur de trafic ». Il est multi-OS et téléchargeable sur le site www.wireshark.com.

Avec *Wireshark*, il est possible de capturer des trames Ethernet en temps réel directement sur les Cartes de communication du terminal, de sauvegarder les résultats de cette capture dans des fichiers qui peuvent être analysés ultérieurement hors ligne. *Wireshark* supporte un très grand nombre de protocoles de communication et de formats de fichiers de capture : Ethernet, ARP, IP, TCP/UDP, HDLC, etc ... libpcap/tcpdump, Sun's snoop/atmsnoop, Lanalyzer, MS Network Monitor, HP-UX nettl, AIX iptrace, Cisco Secure IDS, etc....

Durant ce TP, nous allons :

1. lancer le programme Wireshark,
2. capturer et analyser une trame Ethernet
3. définir des filtres pour la capture et la visualisation des trames
4. Enregistrer le résultat de cette capture dans un fichier

Etape 1 : Lancement de Wireshark

1.1 Démarrez ensuite l'application *Wireshark*. Créez un raccourci sur votre bureau il vous sera bien utile. Voici comment le sniffer se présente.

The screenshot shows the Wireshark interface with the following components:

- Filter:** A text box for applying filters to the packet list.
- Packet List:** A table showing captured packets with columns for No., Time, Source, Destination, Protocol, and Info.
- Packet Details:** A tree view showing the hierarchical structure of the selected packet (Frame 1), including Ethernet II, Internet Protocol, and User Datagram Protocol.
- Packet Bytes:** A pane showing the raw packet data in hexadecimal and ASCII format.

Partie 1

Partie 2

Partie 3

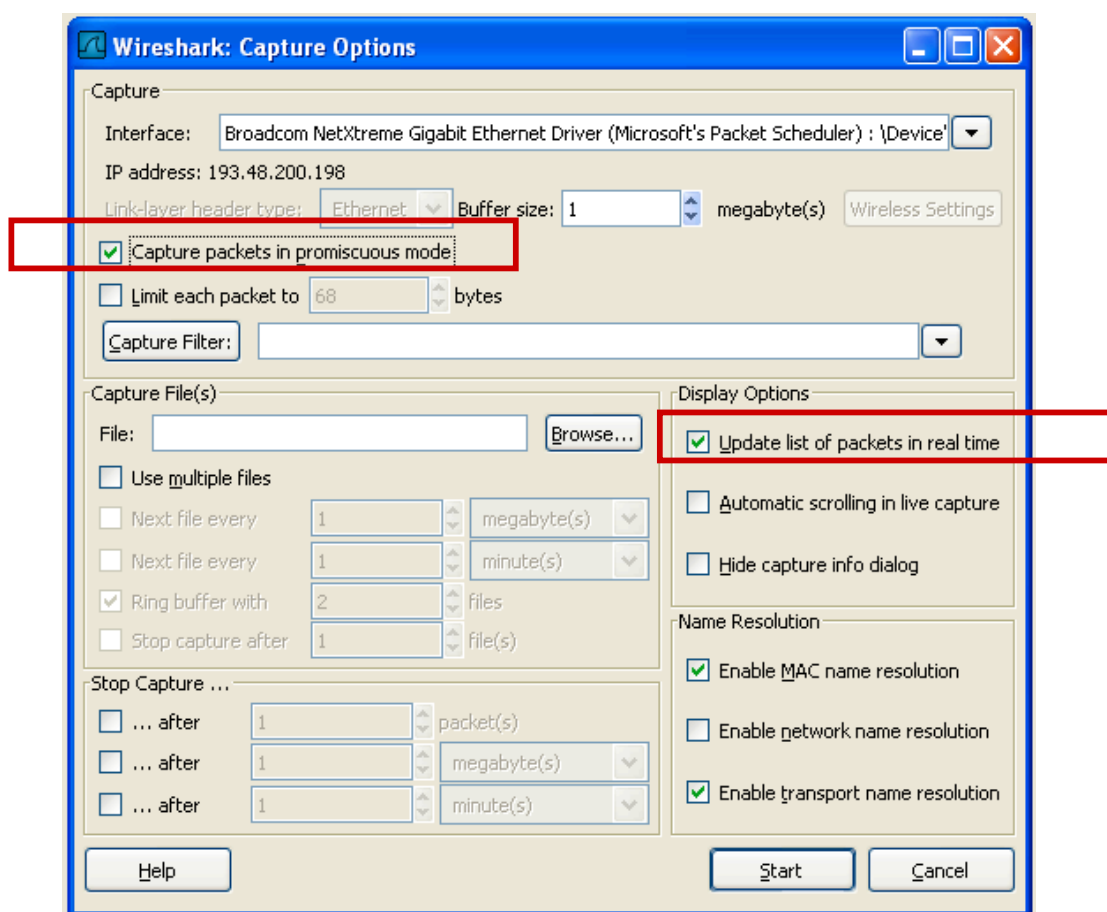
La fenêtre est divisée en trois parties.

1. La **première partie** est de type général, on y trouve des informations de type adresse IP des machines ou encore protocole utilisé lors de l'échange des données.
2. La **deuxième partie** de la fenêtre reprend ici la trame sélectionnée et la détaille soit dans les sept couches du modèles OSI ou dans les quatre couches du modèle IP. Pour plus d'informations à ce sujet des tutoriaux sont disponibles sur le net.
3. La **troisième et dernière partie** est une vision de la trame en codage hexadécimal et ASCII

Nous allons voir maintenant comment capturer les trames sur le réseau sur lequel le sniffer est connecté.

Etape 2 : Capture de trames sur le réseau

Pour capturer les trames sur le réseau, vous devez aller dans le menu "Capture" et cliquez sur "Start". La fenêtre suivante s'ouvre.



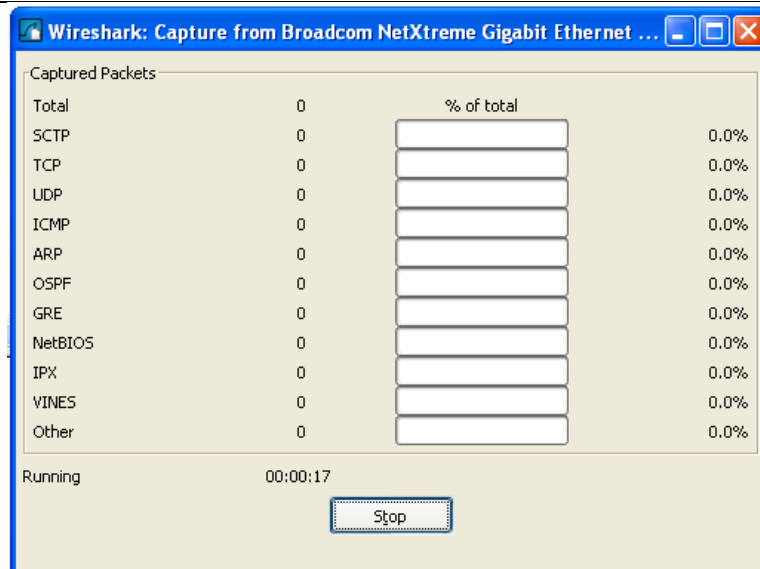
Choisissez l'interface sur laquelle vous voulez "écouter" le trafic. Si vous n'en avez qu'une le choix ne sera pas très difficile.

Par défaut l'espace réservé à la collecte des données est défini à 1MB. Cela devrait être suffisant. Dans le cas contraire augmentez-le.

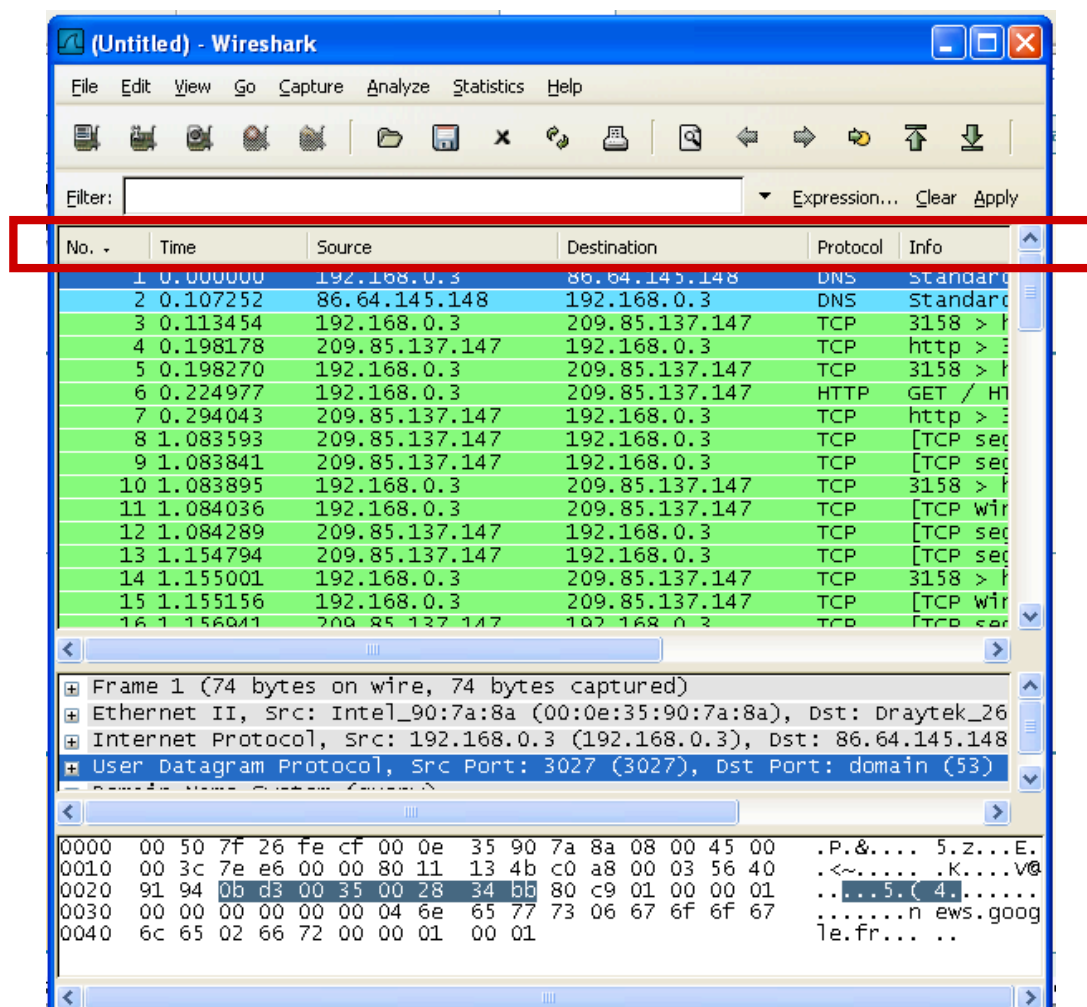
Activer l'option "**Capture packets in promiscuous mode**". Cette option permet à la carte réseau de lire et d'intercepter tout le trafic sur le réseau. Dans le cas contraire celle-ci n'interceptera que les trames qui lui sont destinées et ainsi vous ne verrez pas toutes les trames Multicast et Broadcast.

Laissez le champ "**Capture Filter**" vide dans un premier temps. Nous verrons par la suite comment le remplir. Nous ne toucherons pas non plus aux autres options.

Il ne vous reste plus qu'à démarrer la capture en cliquant sur "OK". La fenêtre suivante s'ouvre.



Capturez environ 30 secondes de trafic entre le poste client et serveur. Puis cliquez sur "Stop". *Wireshark* va alors afficher les trames capturées par votre carte réseau dans un format lisible ci -dessous.



Sur la première partie de cette fenêtre les différentes trames capturées s'affichent et suivant les colonnes nous avons les informations suivantes:

Première colonne : numéro de la trame.

Deuxième colonne : temps écoulé depuis le départ de la capture et l'arrivée de la trame.

Troisième colonne : adresse IP ou nom de la machine émettrice

Quatrième colonne : adresse IP ou nom de la machine réceptrice

Cinquième colonne : protocole utilisé entre les deux machines

Sixième colonne : informations complémentaires

La quantité de données capturées peut vite devenir considérable, d'autant plus que plusieurs communications peuvent être établies en parallèle comme par exemple une connexion à www.google.fr et une autre à www.tplpc.com.

C'est pourquoi nous allons voir comment définir un filtre pour capturer une partie de tout ce que voit la carte réseau.

Etape 3 : Les filtres

Il y a deux sortes de filtres. **Les filtres à la capture et les filtres à l'affichage**. Ces filtres n'ont pas la même syntaxe. Pour Unix la syntaxe des filtres à la capture est la même que les filtres utilisés pour la commande tcpdump. Pour en connaître le format, il faut donc utiliser man tcpdump. Quand aux filtres à l'affichage, la syntaxe est une syntaxe propriétaire à Wireshark. Pour en connaître la syntaxe, il faut utiliser la commande man wireshark. La section présente donne des exemples pour ces deux types de filtres.

1. Filtres de capture

Ne seront conservés que les paquets pour lesquels le filtre est vrai. Les filtres se décomposent en 3 parties :

- le **protocole** à capturer : exemples : ether, fddi, ip, arp, rarp, decnet, lat, sca, moprc, mopdl, tcp ou udp,
- l'identifiant qui peut être src ou dst,
- un champ qui peut être host, net ou port suivi d'une valeur.

Les opérateurs and, or et not peuvent être utilisés pour combiner des filtres.

Filtre	Fonction
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 à destination du port 53
udp port 53 and dst host 172.16.0.1	ne conserve que les paquets UDP en provenance ou à destination du port 53 et 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 à destination de la machine 172.16.0.1 sur le port 80 et en provenance des machines du sous-réseau 172.16.0/24

2. Filtres d'affichage

Les filtres d'affichage sont un peu plus fins que ceux de la capture. Seuls les paquets pour lesquels l'expression du filtre est vraie seront gardés. Les expressions sont basées sur les champs disponibles dans un paquet. Le simple ajout d'un champ veut dire que l'on garde le paquet si ce champ est disponible. Maintenant, on peut aussi utiliser les opérateurs ==, !=, >, <, >= et <= pour comparer les champs avec des valeurs. Les expressions ainsi fabriquées peuvent être combinées avec les opérateurs && (pour un et logique), || (pour un ou logique), ^^ (pour le ou exclusif) et ! Pour la négation. L'usage des parenthèses est possible.

Voici quelques exemples de champs disponibles

Champ	Type	Signification
ip.addr	adresse IPv4	adresse IP source ou destination
ip.dst	adresse IPv4	adresse IP destination
ip.flags.df	booléen	Drapeau IP, ne pas fragmenter
ip.flags.mf	booléen	Drapeau IP, fragments À venir
ip.ttl	entier non signé sur 8 bits	Time to live

nbdgm.src.ip	adresse IPv4	adresse IP source d'un paquet Netbios Datagram
nbdgm.src.port	entier non signé sur 16 bits	port IP source d'un paquet Netbios Datagram
http.request	booléen	requête HTTP
http.response	booléen	réponse HTTP
icmp.code	entier non signé sur 8 bits	numéro du code d'une commande ICMP
icmp.type	entier non signé sur 8 bits	numéro du type d'une commande ICMP
ftp.request	booléen	requête FTP
ftp.request.command	chaîne de caractères	commande FTP
ftp.reponse.data	chaîne de caractères	donnée de transfert FTP
dns.query	booléen	requête DNS
dns.response	booléen	réponse d'une requête DNS

Voici quelques exemples de filtres

Filtre	Signification
ip.addr == 172.16.0.100	tous les paquets IP en provenance ou Ã destination de la machine 172.16.0.100
(ip.addr == 172.16.0.100) && (dns.response)	tous les paquets IP en provenance ou Ã destination de la machine 172.16.0.100 qui sont des réponses Ã des requêtes DNS
(ip.addr >= 172.16.0.100) && (ip.addr <= 172.16.0.123)	tous les paquets IP en provenance ou Ã destination des machines comprises entre l'adresse IP 172.16.0.100 et l'adresse IP 172.16.0.123 (comprises)

3. Comment définir un filtre pour la capture des trames (Capture Filter)

Allez dans le menu "Capture". Puis cliquez sur "Capture Filters".La fenêtre suivante s'ouvre.

Considérons que notre machine a l'adresse IP 192.168.1.33.

Nous voulons capturer uniquement les trames échangées entre celle-ci et la machine avec l'adresse IP 145.200.80.45.

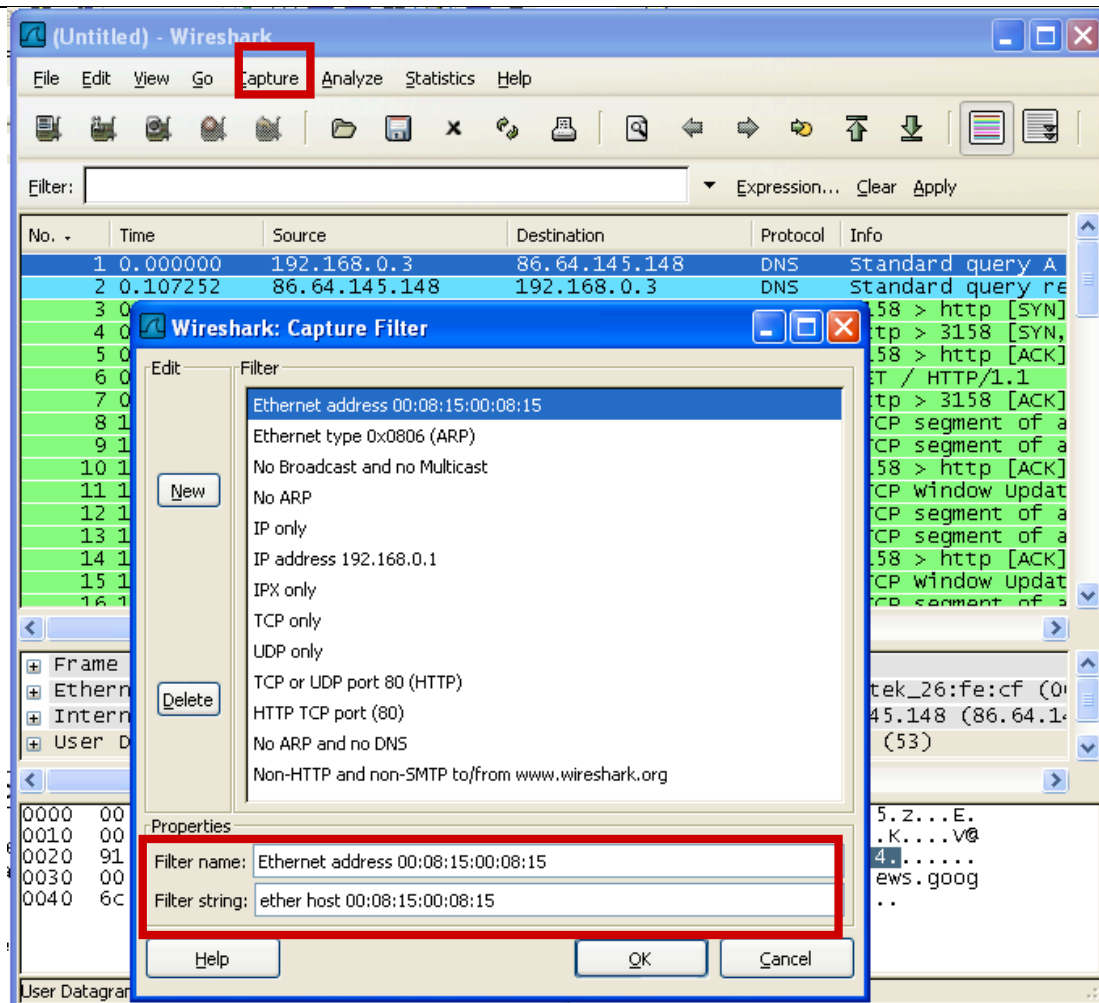
Pour cela cliquez sur "New".

Dans le champ "Filter Name" entrez le nom de votre filtre : mon filtre (par exemple).

Dans le champ "Filter string" entrez la chaîne suivante : host 145.200.80.45. Cliquez maintenant sur "save" et voilà votre filtre est défini vous pouvez cliquez sur "close" pour fermer la fenêtre.

Retournez dans le menu "Capture" et cliquez sur "Start". Reprenez les mêmes options que précédemment. Cliquez sur le bouton "Capture Filter" et sélectionnez votre filtre. Voilà cliquez sur "OK" pour démarrer la capture avec le filtre en question.

Pour plus de détail sur la structure des filtres vous pouvez consulter l'aide en appuyant sur la touche F1 et en allant sur l'onglet "Capture Filter"

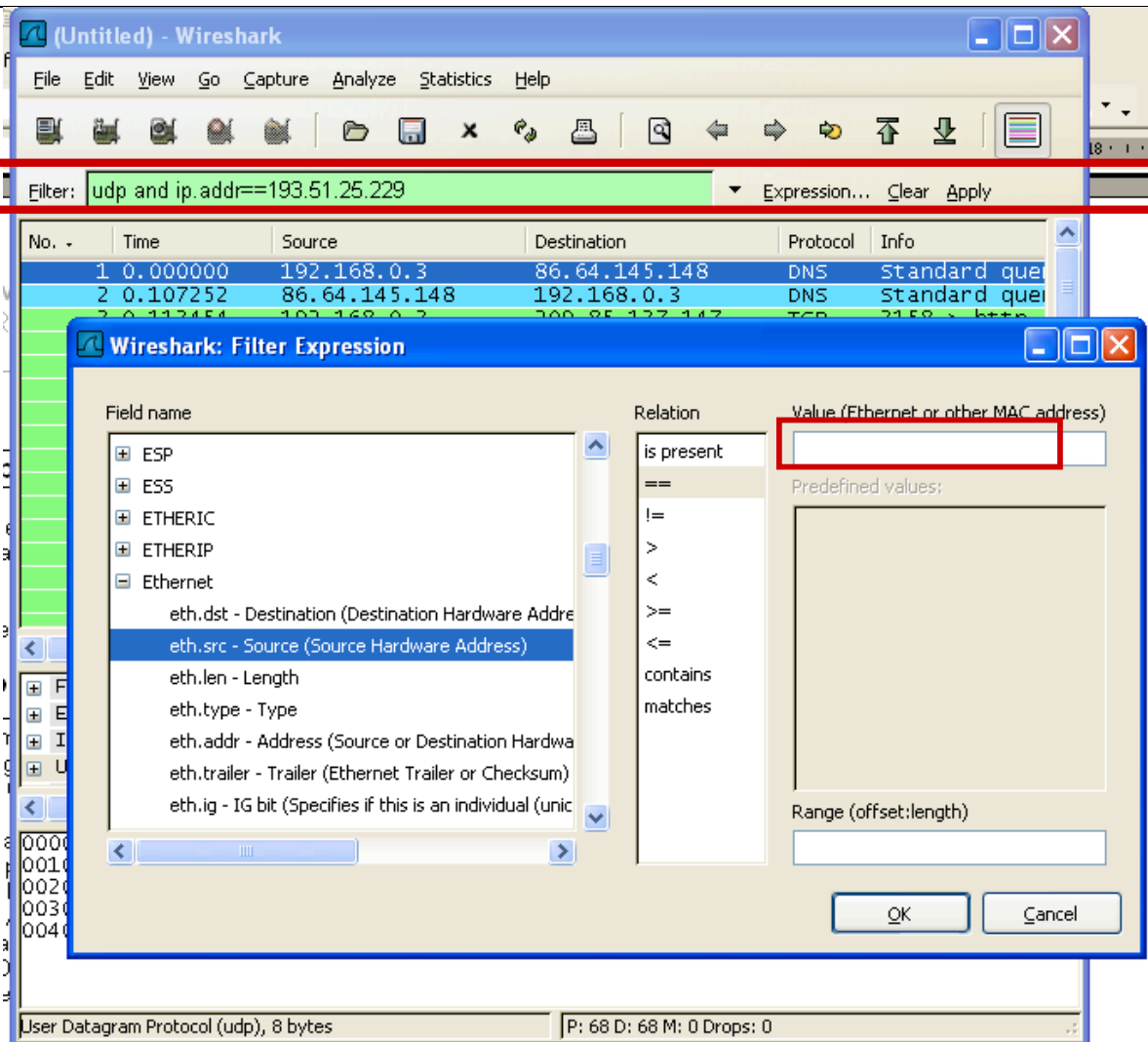


Une autre méthode consiste à capturer toutes les trames dans un premier temps et de filtrer par la suite. L'avantage de cette solution est d'avoir toujours la capture de départ et d'y appliquer par la suite autant de filtres que l'on souhaite. C'est ce que nous allons voir dans le prochain chapitre.

4. Comment définir un filtre pour la visualisation des trames (Display Filter)

Essayons d'appliquer le même filtre que précédemment. Dans un premier temps faites une capture sans appliquer de filtre (reportez vous au premier paragraphe). Stoppez la capture. Allez sur la barre FILTER et sélectionner « EXRESSION ». une fenêtre s'ouvre vous permettant de rédiger des filtres d'affichage. Par exemple on sélectionne le protocole Ethernet et l'adresse source. On tape la chaîne suivante : `eth.src==12:23:45:67:34 5A` et on valide. Voilà le filtre d'affichage est appliqué. Si vous voulez le sauvegarder cliquez sur "Save".

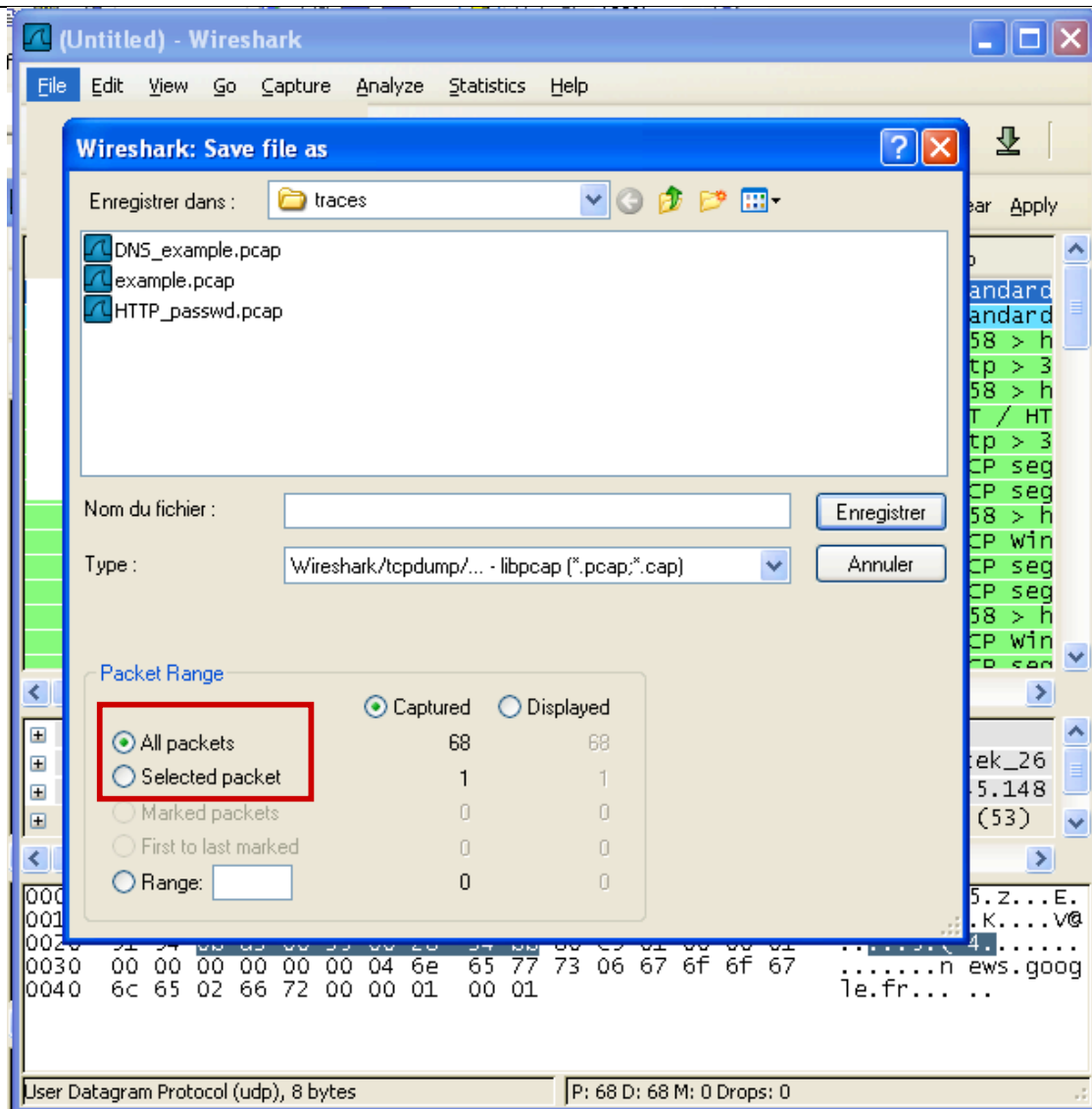
Si maintenant vous voulez l'annuler, effacez la chaîne dans le champ "Filter string" ou cliquez sur « CLEAR ».



Etape 4 : sauvegarde d'un résultat de capture

Pour sauvegarder le résultat d'une capture dans un fichier, il faut sélectionner la commande « Save as » dans le menu « File ». Une fenêtre nous propose de choisir le répertoire et le nom du fichier, ainsi que le format/type de fichier de sauvegarde (conserver le format par défaut libpcap).

Pour n'enregistrer qu'une trame ou une sélection de trames, vous avez à votre disposition ces options dans le menu « Packet Range ».



Etape 5 : Répondre aux questions suivantes :

5.1 taper sur la console de votre terminal la commande « ifconfig » ou « ipconfig » (voir le manuel man pour la syntaxe de la commande ifconfig). Identifier l'adresse MAC (Physique) et l'adresse IP de votre terminal.

Il faut ajouter l'option « /all » ou « -a » pour obtenir l'adresse physique (MAC) :

Adresse MAC est : 00-21-5C-4C-B0-A9

Adresse IP est : 192.168.13.176

5.2 lancer le logiciel Wireshark sur votre interface Ethernet (eth0),

5.3 taper une commande de type « ping » à destination d'une machine voisine et capturer environ 30 secondes de trafic (voir le manuel man pour la syntaxe de la commande ping). Enregistrer le résultat dans un fichier « test ».

Ping est une commande système qui permet de tester la disponibilité d'un hôte (PC, serveur, routeur, imprimante, ...) utilisant le protocole de communication IP.

Ping transmet des requêtes ICMP (ICMP echo), et l'hôte distant doit répondre avec des réponses ICMP (ICMP reply)

C:\Users\Ahmed>ping 192.168.13.1

Pinging 192.168.13.1 with 32 bytes of data:

Reply from 192.168.13.1: bytes=32 time=42ms TTL=255

Ping statistics for 192.168.13.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 42ms, Average = 17ms

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: icmp Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
864	101.657276	192.168.13.130	192.168.13.181	ICMP	Echo (ping) request
865	101.657445	192.168.13.181	192.168.13.130	ICMP	Echo (ping) reply
872	102.602786	192.168.13.130	192.168.13.181	ICMP	Echo (ping) request
873	102.602951	192.168.13.181	192.168.13.130	ICMP	Echo (ping) reply
874	102.960715	192.168.13.130	192.168.13.181	ICMP	Destination unreachable (Port unreachable)
875	103.608370	192.168.13.181	192.168.13.130	ICMP	Echo (ping) request
877	103.608593	192.168.13.181	192.168.13.130	ICMP	Echo (ping) reply
880	104.628668	192.168.13.130	192.168.13.181	ICMP	Echo (ping) request
881	104.628840	192.168.13.181	192.168.13.130	ICMP	Echo (ping) reply
883	105.364558	192.168.13.130	192.168.13.181	ICMP	Echo (ping) request
884	105.364718	192.168.13.181	192.168.13.130	ICMP	Echo (ping) reply
890	106.347915	192.168.13.130	192.168.13.181	ICMP	Echo (ping) request
891	106.348086	192.168.13.181	192.168.13.130	ICMP	Echo (ping) reply
901	107.370166	192.168.13.130	192.168.13.181	ICMP	Echo (ping) request
902	107.370333	192.168.13.181	192.168.13.130	ICMP	Echo (ping) reply
913	108.352801	192.168.13.130	192.168.13.181	ICMP	Echo (ping) request
914	108.352990	192.168.13.181	192.168.13.130	ICMP	Echo (ping) reply

☒ Frame 876 (74 bytes on wire, 74 bytes captured)

☒ Ethernet II, Src: Apple_e2:ce:86 (00:26:08:e2:ce:86), Dst: IntelCor_4c:b0:a9 (00:21:5c:4c:b0:a9)

☒ Internet Protocol, Src: 192.168.13.130 (192.168.13.130), Dst: 192.168.13.181 (192.168.13.181)

☒ Internet Control Message Protocol

Type: 8 (Echo (ping) request)

Code: 0 ()

Checksum: 0x4cfa [correct]

Identifier: 0x0001

Sequence number: 97 (0x0061)

☒ Data (32 bytes)

Data: 6162636465666768696A6B6C6D6E6F707172737475767761...

[Length: 32]

```

0000  00 21 5c 4c b0 a9 00 26 08 e2 ce 86 08 00 45 00  .!\L...&.....E.
0010  00 3c 4e af 00 80 01 4f 8a c0 a8 0d 82 c0 a8  .<N....O.....
0020  00 95 08 00 4c fa 00 01 00 61 61 62 63 64 65 66  .L....aabcd
0030  67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67 68 69                    wabcedfg hi
          
```

File: "C:\Users\Ahmed\AppData\Local\Temp\..."
Packets: 918 Displayed: 65 Marked: 0 Dropped: 0
Profile: Default

Sous unix :
Des dizaines de trames ICMP echo et reply

5.5 Analyser la première trame Ethernet et reporter les valeurs des champs de contrôle de cette trame dans un tableau. Quelle information cette trame transporte t elle ?

Wireshark capture showing ICMP Echo (ping) requests and replies. The filter is set to 'icmp'. The selected packet (Frame 68) is highlighted, and its details are expanded, showing the Ethernet II frame structure. A red box highlights the Ethernet II frame details, specifically the Source and Destination MAC addresses.

No.	Time	Source	Destination	Protocol	Info
2	0.001999	192.168.13.72	192.168.13.176	ICMP	Destination unreachable
68	5.724273	192.168.13.176	192.168.13.1	ICMP	Echo (ping) request
69	5.726266	192.168.13.1	192.168.13.176	ICMP	Echo (ping) reply
79	6.135764	193.55.96.84	192.168.13.176	ICMP	Destination unreachable
83	6.726595	192.168.13.176	192.168.13.1	ICMP	Echo (ping) request
84	6.729064	192.168.13.1	192.168.13.176	ICMP	Echo (ping) reply
86	6.874945	193.55.96.84	192.168.13.176	ICMP	Destination unreachable
98	7.623504	193.55.96.84	192.168.13.176	ICMP	Destination unreachable
106	7.728576	192.168.13.176	192.168.13.1	ICMP	Echo (ping) request
107	7.730459	192.168.13.1	192.168.13.176	ICMP	Echo (ping) reply
117	8.682665	193.55.96.84	192.168.13.176	ICMP	Destination unreachable
118	8.729653	192.168.13.176	192.168.13.1	ICMP	Echo (ping) request
119	8.731688	192.168.13.1	192.168.13.176	ICMP	Echo (ping) reply
129	9.436432	193.55.96.84	192.168.13.176	ICMP	Destination unreachable
142	10.182995	193.55.96.84	192.168.13.176	ICMP	Destination unreachable
149	10.935845	193.55.96.84	192.168.13.176	ICMP	Destination unreachable
154	11.686314	193.55.96.84	192.168.13.176	ICMP	Destination unreachable

Frame 68 (74 bytes on wire (58 bytes captured) on interface 0:00:00:00:00:00):
 Ethernet II, Src: IntelCor_4c:b0:a9 (00:21:5c:4c:b0:a9), Dst: Netgear_ff:d2:ba (00:09:5b:ff:d2:ba)
 Destination: Netgear_ff:d2:ba (00:09:5b:ff:d2:ba)
 Source: IntelCor_4c:b0:a9 (00:21:5c:4c:b0:a9)
 Type: IP (0x0800)
 Internet Protocol, Src: 192.168.13.176 (192.168.13.176), Dst: 192.168.13.1 (192.168.13.1)
 Internet Control Message Protocol

0000 00 09 5b ff d2 ba 00 21 5c 4c b0 a9 08 00 45 00 ...! \L...E.
 0010 00 3c 21 c1 00 00 80 01 7c fe c0 a8 0d b0 c0 a8 .<!. |.
 0020 0d 01 08 00 4d 4c 00 01 00 0f 61 62 63 64 65 66 ML abcdef
 0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmn opqrstuv
 0040 77 61 62 63 64 65 66 67 68 69 wabcdefg hi

Destination Hardware Address (eth.dst), 6 bytes: 00:09:5b:ff:d2:ba
 Packets: 968 Displayed: 45 Marked: 0 Dropped: 0
 Profile: Default

Ils est possibles d'identifier la source de ce test ping : en retrouvant son adresse physique et IP (00:09:5b:ff:d2:ba et 192.168.13.176)

5.6 Recherchez sur Internet le document RFC 1700. Quelle information mentionne t il en relation avec la trame Ethernet ?

Un RFC (Request For Comment) est un document de spécification des règles de communication de l'Internet (Intranet).

Le RFC de n° 1700 contient la liste des codes de protocoles et les n° de ports réservés pour des usages particuliers.

Le RFC 1700 a été remplacé par le RFC 3232

5.7 Au moyen des filtres d'affichage, sélectionner uniquement les trames dont l'émetteur est votre poste client (sur la base de son adresse Ethernet).

Filtre : eth.src == 00-21-5C-4C-B0-A9

The screenshot shows the Wireshark interface with a packet capture. The filter expression dialog box is open, showing the field name "eth.dst - Destination (Destination Hardware Address)", the relation "is present", and the value "192.168.13.130". The packet list shows several ICMP Echo (ping) requests and replies. The packet details pane shows the selected packet (No. 876) with its Ethernet II, Internet Protocol, and Internet Control Message Protocol (ICMP) fields.

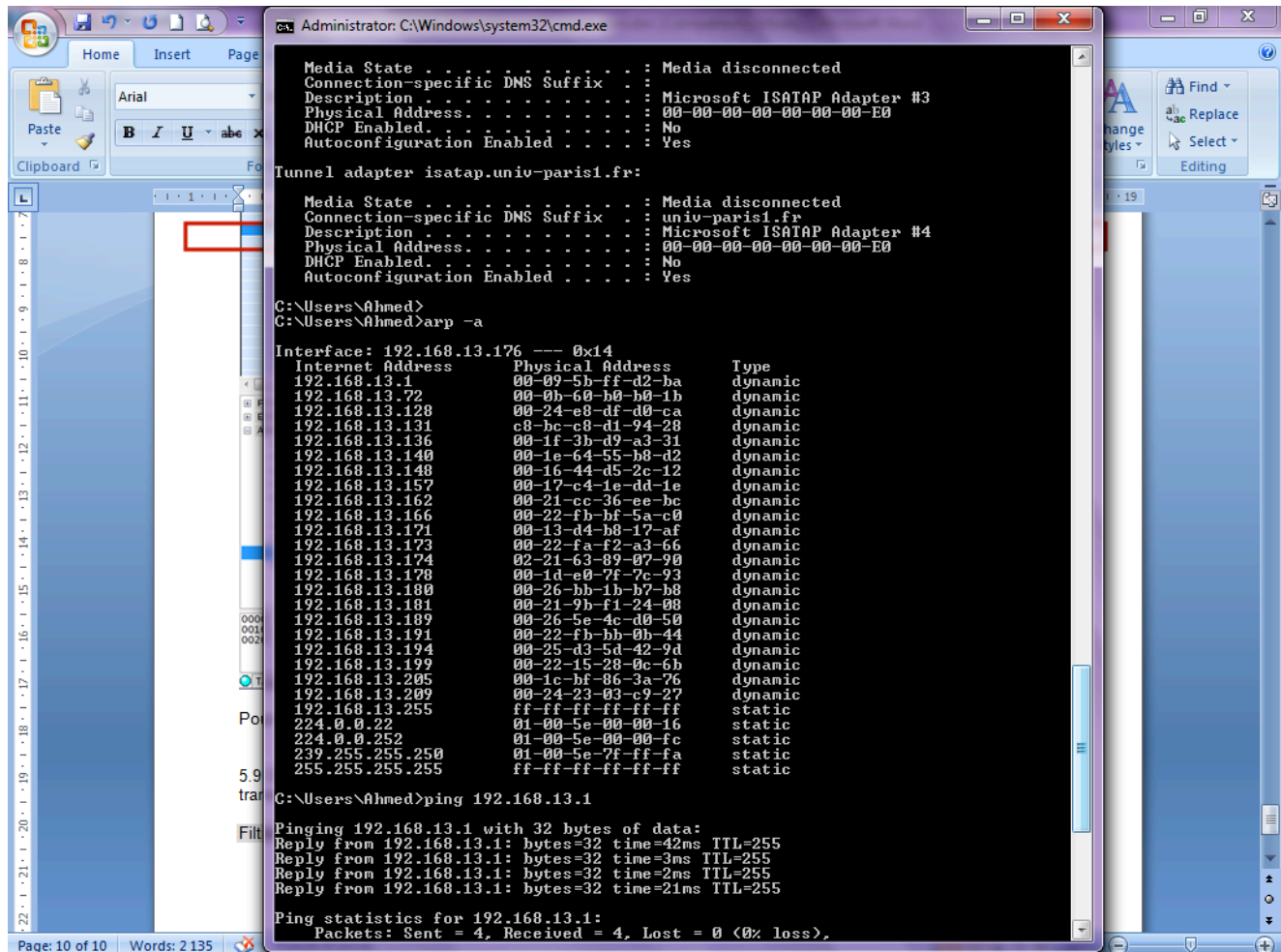
The screenshot shows the Wireshark interface with a packet capture. The filter expression is set to "arp". The packet list shows several ARP requests and replies. The packet details pane shows the selected packet (No. 131) with its Ethernet II, Internet Protocol, and Address Resolution Protocol (ARP) fields. The ARP field shows the target MAC address as "00:00:00:00:00:00" and the target IP address as "192.168.13.72".

5.8 A quoi servent les trames ARP ?

ARP signifie Address Resolution Protocol. Ce logiciel de communication permet de trouver l'adresse Physique (MAC) d'un hôte sur le même réseau local (ex. DNS, routeur par défaut, un serveur web local, une imprimante réseau ...) connaissant son adresse IP.

L'ordinateur envoie une requête ARP et il reçoit une réponse ARP.

Pour visualiser toutes les adresses physiques (MAC) déjà obtenues, il suffit de taper la commande « arp -a ».



```
Administrator: C:\Windows\system32\cmd.exe

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . : univ-paris1.fr
Physical Address. . . . . : 00-00-00-00-00-00-E0
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes

Tunnel adapter {isatap.univ-paris1.fr}:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . : univ-paris1.fr
Description . . . . . : Microsoft ISATAP Adapter #4
Physical Address. . . . . : 00-00-00-00-00-00-E0
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes

C:\Users\Ahmed>
C:\Users\Ahmed>arp -a

Interface: 192.168.13.176 --- 0x14
Internet Address      Physical Address      Type
192.168.13.1          00-09-5b-ff-d2-ba     dynamic
192.168.13.72         00-0b-60-b0-b0-1b     dynamic
192.168.13.128        00-24-e8-df-d0-ca     dynamic
192.168.13.131        c8-bc-c8-d1-94-28     dynamic
192.168.13.136        00-1f-3b-d9-a3-31     dynamic
192.168.13.148        00-1e-64-55-b8-d2     dynamic
192.168.13.149        00-16-44-d5-2e-12     dynamic
192.168.13.157        00-17-e4-1e-dd-1e     dynamic
192.168.13.162        00-21-cc-36-ee-bc     dynamic
192.168.13.166        00-22-fb-bf-5a-c0     dynamic
192.168.13.171        00-13-d4-b8-17-af     dynamic
192.168.13.173        00-22-fa-f2-a3-66     dynamic
192.168.13.174        02-21-63-89-07-90     dynamic
192.168.13.178        00-1d-e0-7f-7c-93     dynamic
192.168.13.180        00-26-bb-1b-b7-b8     dynamic
192.168.13.181        00-21-9b-f1-24-08     dynamic
192.168.13.189        00-26-5e-4c-d0-50     dynamic
192.168.13.191        00-22-fb-bb-0b-44     dynamic
192.168.13.194        00-25-d3-5d-42-9d     dynamic
192.168.13.199        00-22-15-28-0c-6b     dynamic
192.168.13.205        00-1c-bf-86-3a-76     dynamic
192.168.13.209        00-24-23-03-c9-27     dynamic
192.168.13.255        ff-ff-ff-ff-ff-ff     static
224.0.0.22            01-00-5e-00-00-16     static
224.0.0.252           01-00-5e-00-00-fc     static
239.255.255.255       01-00-5e-7f-ff-fa     static
255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\Users\Ahmed>ping 192.168.13.1

Pinging 192.168.13.1 with 32 bytes of data:
Reply from 192.168.13.1: bytes=32 time=42ms TTL=255
Reply from 192.168.13.1: bytes=32 time=3ms TTL=255
Reply from 192.168.13.1: bytes=32 time=2ms TTL=255
Reply from 192.168.13.1: bytes=32 time=21ms TTL=255

Ping statistics for 192.168.13.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

5.9 Décrivez la procédure (commandes systèmes, filtres wireshark) permettant de capturer et de filtrer les trames Ethernet transportant uniquement un paquet ARP.

Filtre : arp