

Wireshark

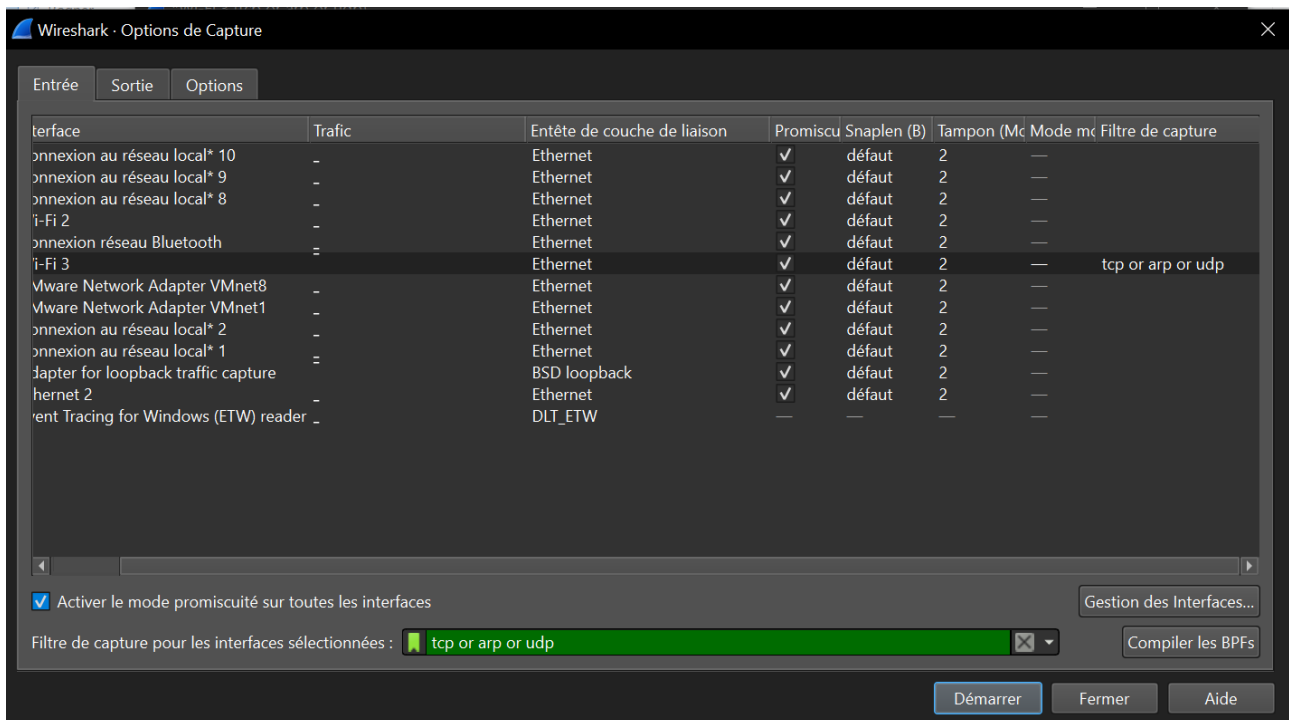
PARTIE 1 :

Quelle est la différence entre une trame et un paquet ? Qu'est-ce que le format pcap/pcapng ?

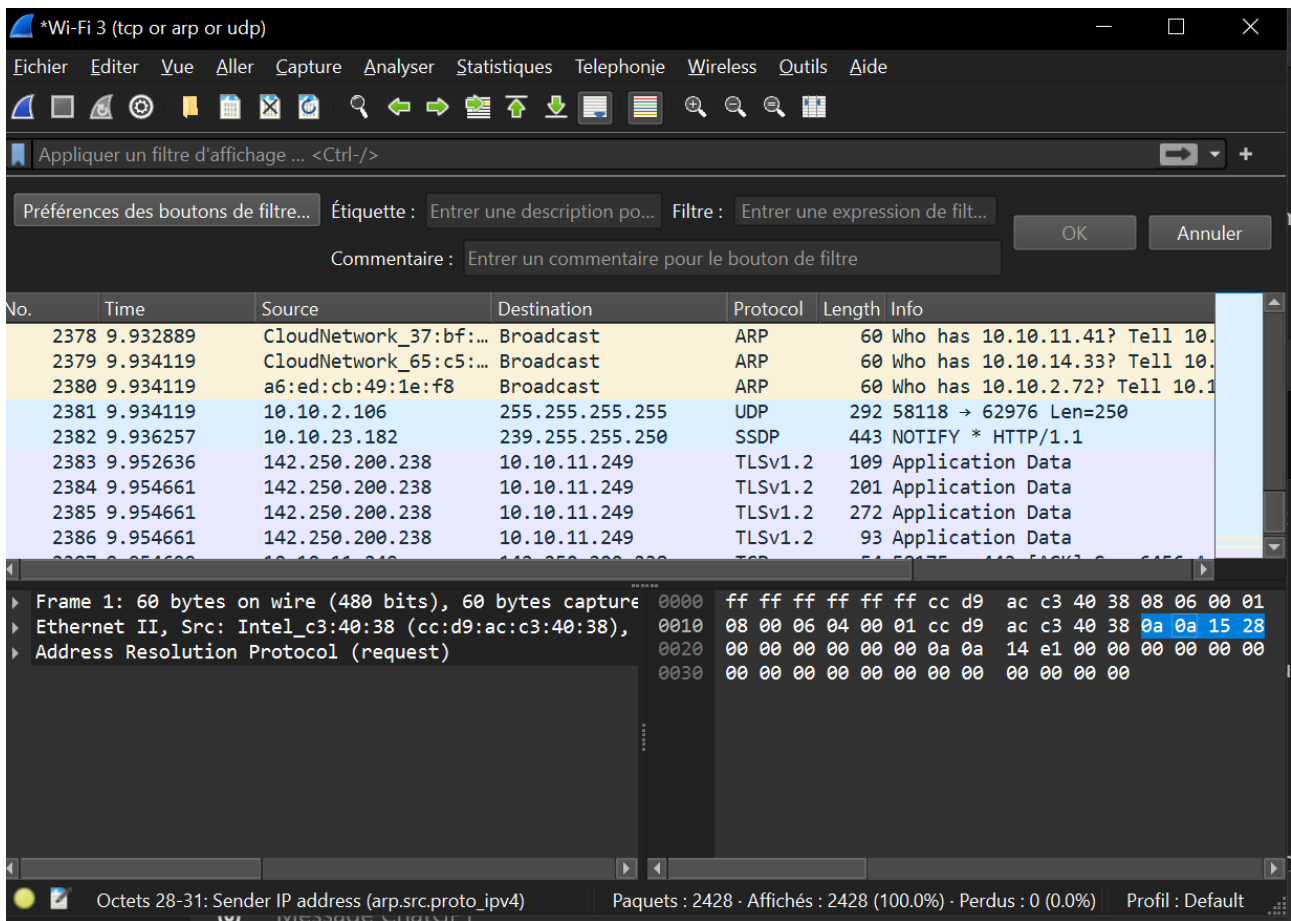
Une trame est utilisée pour envoyer des données entre un seul réseau, tandis qu'un paquet est utilisé pour envoyer des données d'un réseau à un autre, puis vers un périphérique spécifique sur ce réseau

Un fichier PCAP est un fichier issu d'un sniffe de flux réseaux (par exemple avec **Wireshark** ou **TCPDump**), ce sniff réseaux ayant été traité avec la librairie **libpcap** qui produit justement un enregistrement au format **PCAP** (ou **PCAPNG** parfois). Ces fichiers peuvent alors être transportés, stockés et rouverts plus tard, on voit alors tout l'intérêt de savoir les manipuler avec précisions.

Nous allons installer Wireshark sur Windows et nous allons analyser l'interface connecté à Alcasar et nous allons appliquer les filtres pour analyser les paquets ARP, UDP et TCP

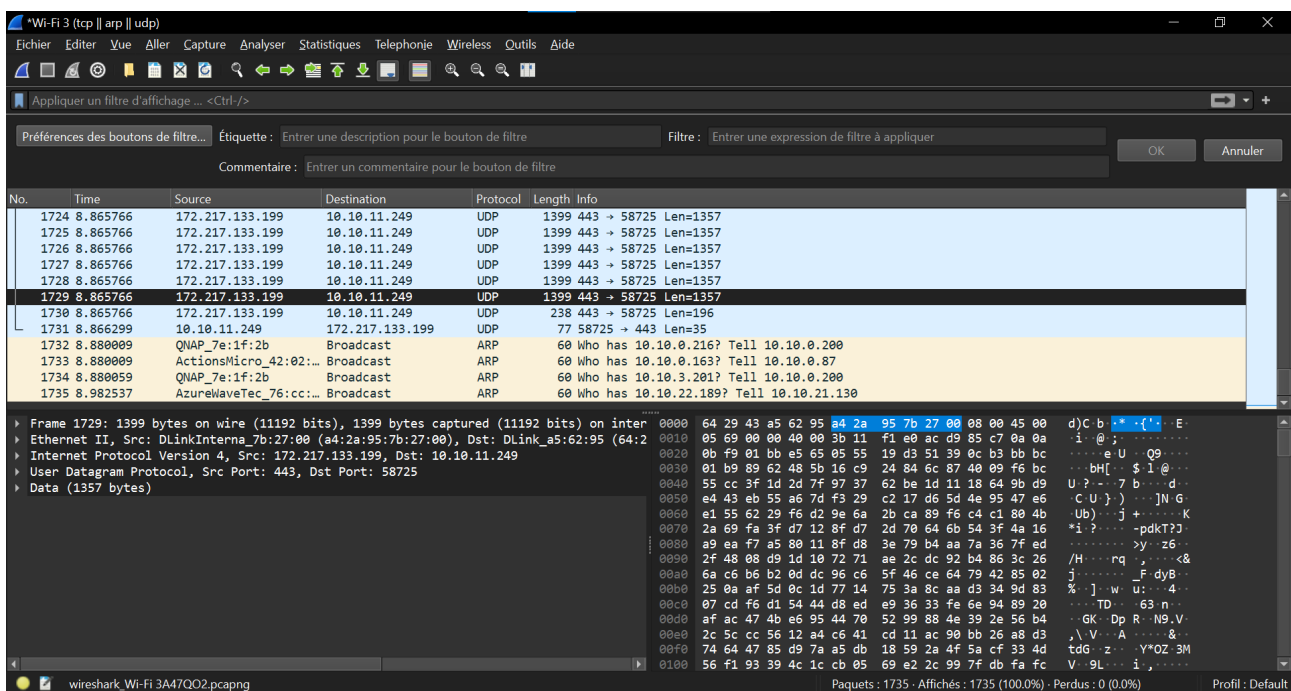


Et voici ce que cela donne

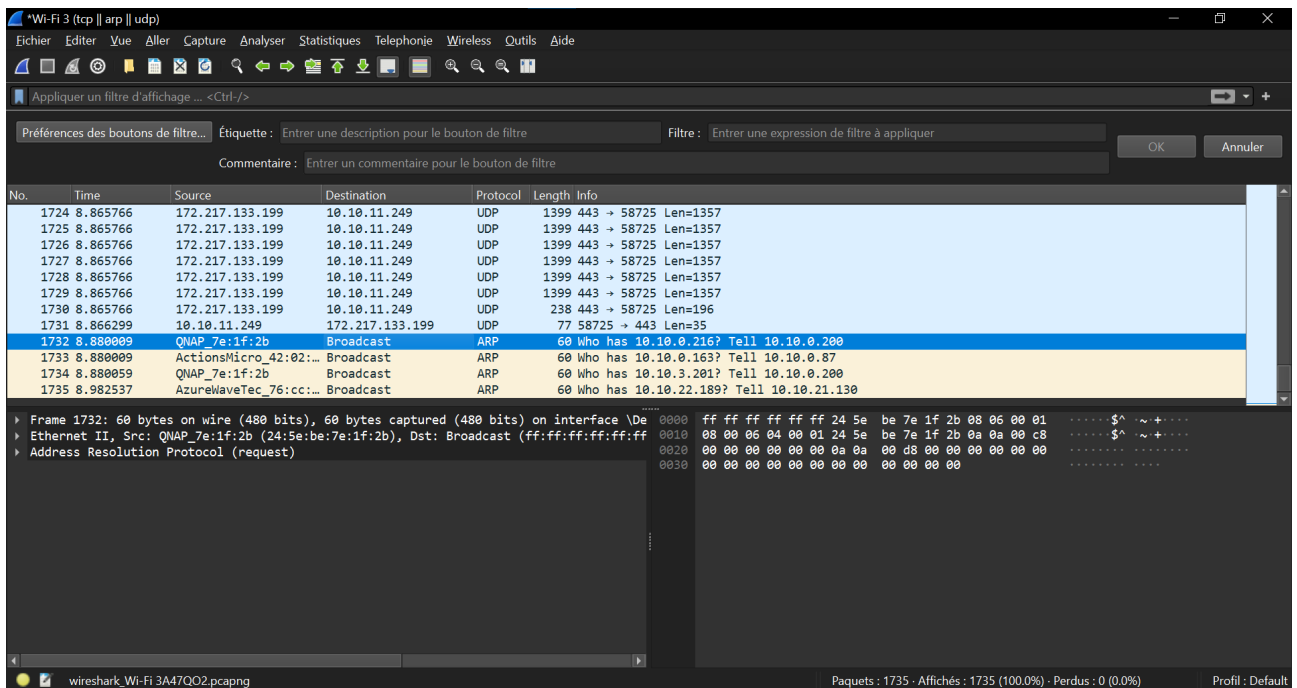


Et nous voyons ici une trame :

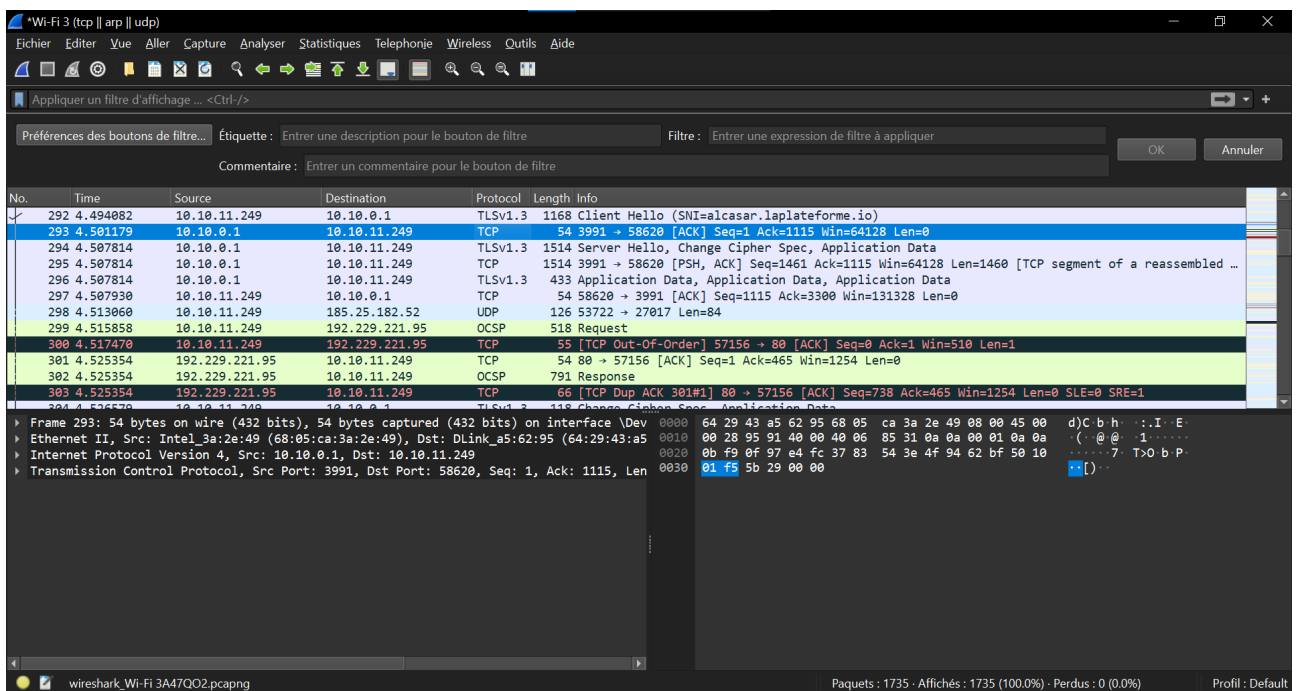
UDP :



ARP :

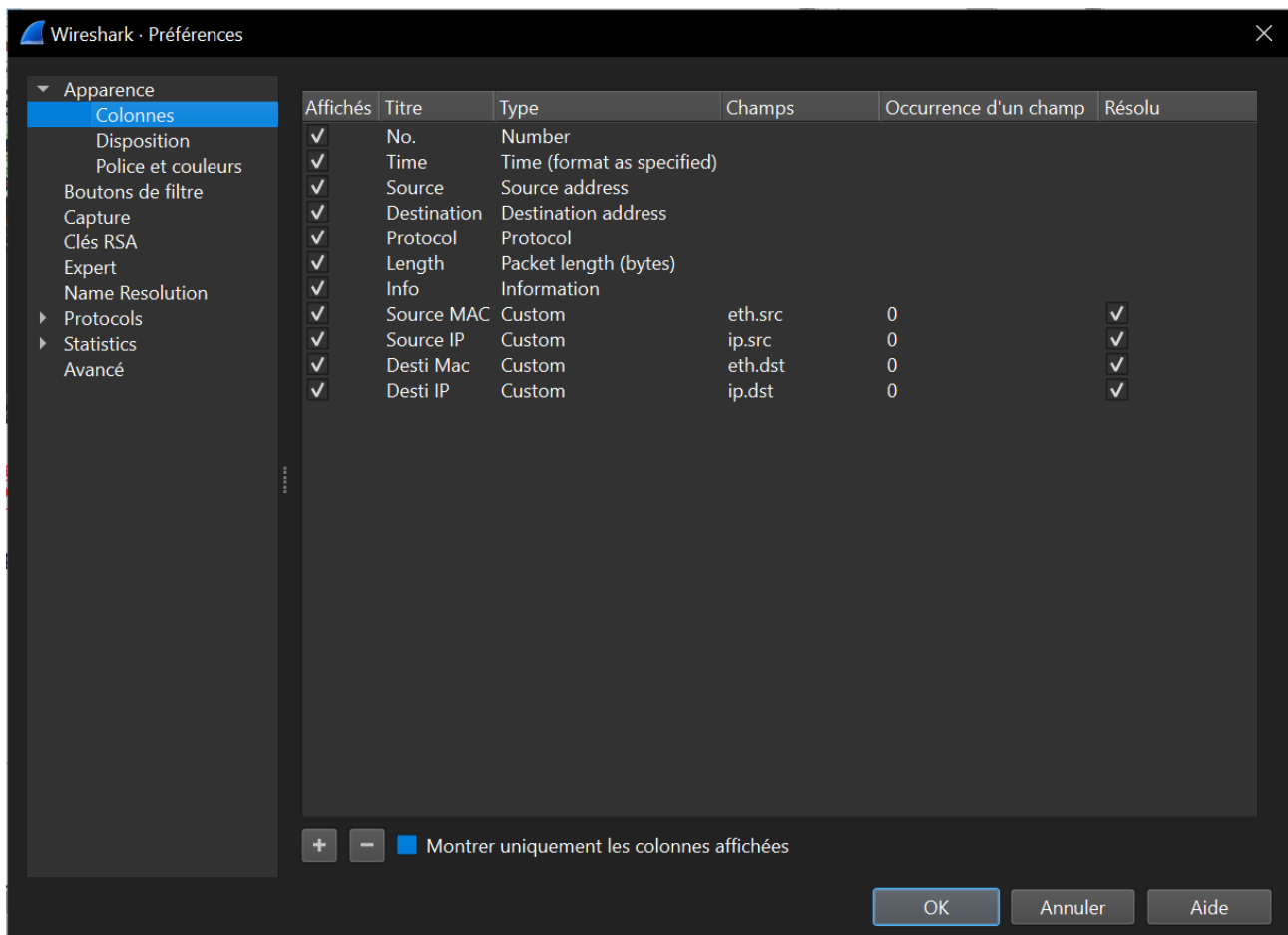


TCP :

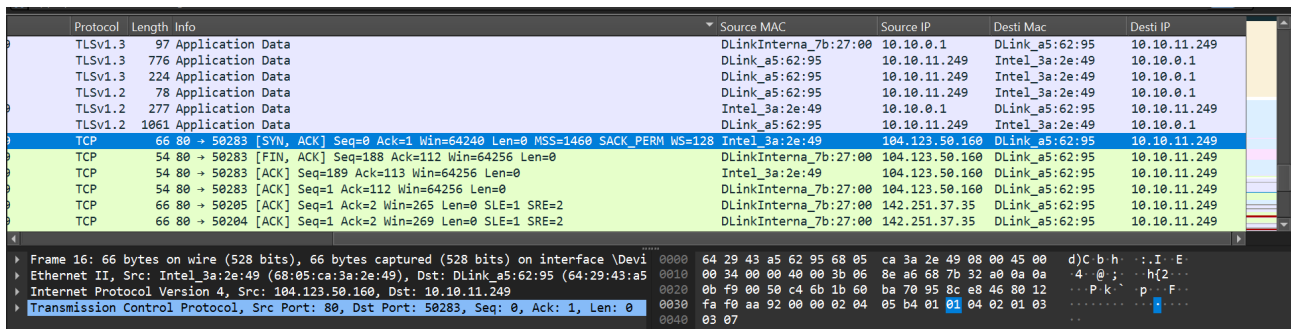


Quelles sont les adresses MAC sources, les IP sources et les adresses MAC sources, les IP destinations des données capturées ?

D'abord, pour pouvoir voir toute ces données, il nous faut pouvoir les ajouter sur nos colonnes d'affichage, nous allons aller dans l'onglet éditer et ensuite dans préférences :



Maintenant, on peut voir ces infos là avec les paquets TCP par exemple



Référez d'autres trames ou paquets circulants sur le réseau. Identifiez leurs protocoles et leur fonction.

No.	Time	Source	Destination	Protocol	Length	Info	Source MAC
92	0.916029	fe80::1cb1:8b26:4c1...	ff02::fb	MDNS	183	Standard query 0x0000 ANY MacBook-Pro-de-Ekaterina.local, "QM" question	AA... Apple_b6:31:1e
93	0.916177	fe80::3a36:88c8:37d...	ff02::16	ICMPv6	90	Multicast Listener Report Message v2	Chongqingfug_9d:a
94	0.917159	fe80::3a36:88c8:37d...	ff02::16	ICMPv6	90	Multicast Listener Report Message v2	Chongqingfug_9d:a
95	0.917159	fe80::3a36:88c8:37d...	ff02::16	ICMPv6	90	Multicast Listener Report Message v2	Chongqingfug_9d:a
96	0.917159	fe80::3a36:88c8:37d...	ff02::16	ICMPv6	90	Multicast Listener Report Message v2	Chongqingfug_9d:a
97	0.917159	10.10.33.171	224.0.0.251	MDNS	81	Standard query 0x0000 ANY LAPTOP-IOSVPLRH.local, "QM" question	Chongqingfug_9d:a
98	0.918252	fe80::3a36:88c8:37d...	ff02::fb	MDNS	101	Standard query 0x0000 ANY LAPTOP-IOSVPLRH.local, "QM" question	Chongqingfug_9d:a
99	0.918252	fe80::3a36:88c8:37d...	ff02::1:3	LLMNR	95	Standard query 0xe699 ANY LAPTOP-IOSVPLRH	Chongqingfug_9d:a
100	0.918365	10.10.33.171	224.0.0.252	LLMNR	75	Standard query 0xe699 ANY LAPTOP-IOSVPLRH	Chongqingfug_9d:a
101	0.918365	10.10.33.171	224.0.0.251	MDNS	81	Standard query 0x0000 ANY LAPTOP-IOSVPLRH.local, "QM" question	Chongqingfug_9d:a
102	0.919453	fe80::3a36:88c8:37d...	ff02::fb	MDNS	101	Standard query 0x0000 ANY LAPTOP-IOSVPLRH.local, "QM" question	Chongqingfug_9d:a
103	0.919453	10.10.33.171	224.0.0.251	MDNS	119	Standard query response 0x0000 AAAA fe80::3a36:88c8:37d6:96fe A 10.10.33.1...	Chongqingfug_9d:a

Il y a par exemple des trames avec les protocoles MDNS LLMNR et ICPMv6

Spécifications des Formats des Messages ARP, UDP et TCP :

ARP (Address Resolution Protocol)

Le message ARP est utilisé pour mapper une adresse IP à une adresse MAC. Le format d'un message ARP est le suivant :

- **Hardware Type (2 bytes):** Typiquement 1 pour Ethernet.
- **Protocol Type (2 bytes):** Typiquement 0x0800 pour IPv4.
- **Hardware Address Length (1 byte):** Taille de l'adresse MAC (6).
- **Protocol Address Length (1 byte):** Taille de l'adresse IP (4).
- **Operation (2 bytes):** 1 pour la requête, 2 pour la réponse.
- **Sender Hardware Address (6 bytes):** Adresse MAC de l'expéditeur.
- **Sender Protocol Address (4 bytes):** Adresse IP de l'expéditeur.
- **Target Hardware Address (6 bytes):** Adresse MAC du destinataire (souvent 0 pour une requête).
- **Target Protocol Address (4 bytes):** Adresse IP du destinataire.

UDP (User Datagram Protocol)

Le format d'un segment UDP est le suivant :

- **Source Port (2 bytes):** Port source.
- **Destination Port (2 bytes):** Port destination.
- **Length (2 bytes):** Longueur totale du segment UDP (en-tête + données).
- **Checksum (2 bytes):** Vérifie l'intégrité des données.

TCP (Transmission Control Protocol)

Le format d'un segment TCP est le suivant :

- **Source Port (2 bytes):** Port source.
- **Destination Port (2 bytes):** Port destination.
- **Sequence Number (4 bytes):** Numéro de séquence.
- **Acknowledgment Number (4 bytes):** Numéro d'accusé de réception (si ACK est défini).
- **Data Offset (4 bits):** Longueur de l'en-tête TCP.
- **Reserved (3 bits):** Réserve pour une utilisation future.
- **Flags (9 bits):** URG, ACK, PSH, RST, SYN, FIN.
- **Window Size (2 bytes):** Taille de la fenêtre de réception.
- **Checksum (2 bytes):** Vérifie l'intégrité des données.
- **Urgent Pointer (2 bytes):** Pointe vers les données urgentes (si URG est défini).
- **Options (variable):** Options TCP (si présentes).
- **Data (variable):** Données encapsulées.

Une capture ARP en hexadécimal (En bas à droite)

No.	Time	Source	Destination	Protocol	Length	Info	Source MAC
68	0.716073	10.10.32.50	224.0.0.251	MDNS	464	Standard query response 0x0000 PTR MacBook Pro Trystan_companion-link_tc...	Apple_00:a9:95
69	0.717485	fe80::437:ae3d:4630...	ff02::fb	MDNS	484	Standard query response 0x0000 PTR MacBook Pro Trystan_companion-link_tc...	Apple_00:a9:95
70	0.746775	10.10.28.216	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1	Intel_88:f9:af
71	0.812074	Apple_46:78:05	Broadcast	ARP	60	Who has 10.10.33.97? Tell 10.10.32.18	Apple_46:78:05
72	0.812074	TpLinkPte_88:32:2e	Broadcast	ARP	60	Who has 10.10.31.179? Tell 10.10.32.202	TpLinkPte_88:32:2e
73	0.812074	AzureWaveTec_3e:96...	Broadcast	ARP	60	Who has 10.10.29.129? Tell 10.10.33.69	AzureWaveTec_3e:96
74	0.812074	CloudNetwork_70:5d...	Broadcast	ARP	60	Who has 10.10.33.69? Tell 10.10.33.57	CloudNetwork_70:5d
75	0.813170	fe80::bbfc:1686:266...	ff02::16	ICMPv6	98	Multicast Listener Report Message v2	CloudNetwork_f7:3
76	0.813170	fe80::bbfc:1686:266...	ff02::16	ICMPv6	98	Multicast Listener Report Message v2	CloudNetwork_f7:3
77	0.813170	fe80::bbfc:1686:266...	ff02::16	ICMPv6	98	Multicast Listener Report Message v2	CloudNetwork_f7:3
78	0.813170	10.10.24.200	224.0.0.251	MDNS	81	Standard query 0x0000 ANY DESKTOP-CIO3E3P.local, "QM" question	CloudNetwork_f7:3
79	0.814487	fe80::bbfc:1686:266...	ff02::fb	MDNS	101	Standard query 0x0000 ANY DESKTOP-CIO3E3P.local, "QM" question	CloudNetwork_f7:3

> Frame 71: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Devic...
 > Ethernet II, Src: Apple_46:78:05 (a8:8f:d9:46:78:05), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 > Address Resolution Protocol (request)

Une capture TCP en hexadécimal :

No.	Time	Source	Destination	Protocol	Length	Info	Source MAC
155	1.471567	10.10.0.1	10.10.11.249	DNS	229	Standard query response 0xb6e4 A v10.events.data.microsoft.com CNAME win-g...	Intel_3a:2e:49
156	1.472191	10.10.11.249	13.89.178.27	TCP	66	50059 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM	DLink_a5:62:95
157	1.475544	10.10.32.87	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1	AzureWaveTec_de:1
158	1.528908	Intel_d5:20:55	Broadcast	ARP	60	Who has 10.10.34.45? Tell 10.10.30.7	Intel_d5:20:55
159	1.528908	ActionsMicro_18:97...	Broadcast	ARP	60	Who has 10.10.32.87? Tell 10.10.2.73	ActionsMicro_18:97
160	1.528936	CloudNetwork_9b:d5...	Broadcast	ARP	60	Who has 10.10.29.22? Tell 10.10.34.29	CloudNetwork_9b:d5
161	1.595389	13.89.178.27	10.10.11.249	TCP	66	443 → 50059 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 WS=256 SACK_PERM	Intel_3a:2e:49
162	1.595453	10.10.11.249	13.89.178.27	TCP	54	50059 → 443 [ACK] Seq=1 Ack=1 Win=132352 Len=0	DLink_a5:62:95
163	1.595784	10.10.11.249	13.89.178.27	TLSv1.2	268	Client Hello (SNI=v10.events.data.microsoft.com)	DLink_a5:62:95
164	1.631315	AzureWaveTec_15:2f...	Broadcast	ARP	60	Who has 10.10.30.47? Tell 10.10.33.114	AzureWaveTec_15:2f
165	1.631315	CloudNetwork_70:68...	Broadcast	ARP	60	Who has 10.10.26.122? Tell 10.10.32.217	CloudNetwork_70:68
166	1.631315	CloudNetwork_58:1c...	Broadcast	ARP	60	Who has 10.10.29.54? Tell 10.10.32.191	CloudNetwork_58:1c

> Frame 162: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Devic...
 > Ethernet II, Src: DLink_a5:62:95 (64:29:43:a5:62:95), Dst: Intel_3a:2e:49 (68:05:ca:3a...)
 > Internet Protocol Version 4, Src: 10.10.11.249, Dst: 13.89.178.27
 > Transmission Control Protocol, Src Port: 50059, Dst Port: 443, Seq: 1, Ack: 1, Len: 0

Et une capture UDP en hexadécimal :

No.	Time	Source	Destination	Protocol	Length	Info	Source MAC
314	2.655281	CloudNetwork_58:1c...	Broadcast	ARP	60	Who has 10.10.30.38? Tell 10.10.32.191	CloudNetwork_58:1c
315	2.655311	Apple_64:12:4a	Broadcast	ARP	60	Who has 10.10.33.97? Tell 10.10.32.169	Apple_64:12:4a
316	2.655311	ActionsMicro_4d:fa...	Broadcast	ARP	60	Who has 10.10.24.55? Tell 10.10.0.88	ActionsMicro_4d:fa
317	2.656527	ActionsMicro_a9:f8...	Broadcast	ARP	60	Who has 10.10.24.55? Tell 10.10.30.123	ActionsMicro_a9:f8
318	2.656527	TpLinkPte_88:29:05	Broadcast	ARP	60	Who has 10.10.26.134? Tell 10.10.32.232	TpLinkPte_88:29:05
319	2.656555	TpLinkPte_88:2d:52	Broadcast	ARP	60	Who has 10.10.24.153? Tell 10.10.26.157	TpLinkPte_88:2d:52
320	2.656555	10.10.34.15	255.255.255.255	UDP	162	6537 → 6537 Len=120	HuiZhouGaosh_1b:b
321	2.673398	10.10.34.61	239.255.255.250	SSDP	142	M-SEARCH * HTTP/1.1	DLinkInterna_0d:4
322	2.673398	10.10.34.61	239.255.255.250	SSDP	142	M-SEARCH * HTTP/1.1	DLinkInterna_0d:4
323	2.683568	10.10.0.88	239.255.255.250	SSDP	270	HTTP/1.1 200 OK	DLinkInterna_0d:4
324	2.683568	10.10.32.50	239.255.255.250	SSDP	218	M-SEARCH * HTTP/1.1	DLinkInterna_0d:4
325	2.683568	10.10.30.123	239.255.255.250	SSDP	272	HTTP/1.1 200 OK	DLinkInterna_0d:4

> Frame 320: 162 bytes on wire (1296 bits), 162 bytes captured (1296 bits) on interface...
 > Ethernet II, Src: HuiZhouGaosh_1b:bf:f8 (4c:50:dd:1b:bf:f8), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 > Internet Protocol Version 4, Src: 10.10.34.15, Dst: 255.255.255.255
 > User Datagram Protocol, Src Port: 6537, Dst Port: 6537
 > Data (120 bytes)

Les paquets surlignés en noir correspondent aux paquets du poste client vers le serveur Web

Les paquets surlignés en vert correspondent aux paquets du serveur Web vers le poste client

18	2.377095	54.192.65.12	10.10.11.249	TCP	66	80 → 49828 [ACK] Seq=1 Ack=2 Win=131 Len=0 SLE=1 SRE=2	Intel_3a:2e:49
19	2.464007	10.10.11.249	192.229.221.95	TCP	55	49632 → 80 [ACK] Seq=1 Ack=1 Win=513 Len=1	DLink_a5:62:95
20	2.472967	192.229.221.95	10.10.11.249	TCP	66	80 → 49632 [ACK] Seq=1 Ack=2 Win=233 Len=0 SLE=1 SRE=2	Intel_3a:2e:49
21	3.104665	10.10.11.249	95.101.110.185	TCP	66	49859 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM	DLink_a5:62:95
22	3.112594	95.101.110.185	10.10.11.249	TCP	66	80 → 49859 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128	Intel_3a:2e:49
23	3.112639	10.10.11.249	95.101.110.185	TCP	54	49859 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0	DLink_a5:62:95
24	3.112837	10.10.11.249	95.101.110.185	HTTP	165	GET /connecttest.txt HTTP/1.1	DLink_a5:62:95
25	3.119216	95.101.110.185	10.10.11.249	TCP	66	[TCP Out-Of-Order] 80 → 49859 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM WS=128	Intel_3a:2e:49
26	3.119240	10.10.11.249	95.101.110.185	TCP	66	[TCP Dup ACK #23] 49859 → 80 [ACK] Seq=112 Ack=1 Win=131328 Len=0 SLE=0 SRE=0	DLink_a5:62:95
27	3.120182	95.101.110.185	10.10.11.249	TCP	54	80 → 49859 [ACK] Seq=1 Ack=112 Win=64256 Len=0	Intel_3a:2e:49
28	3.121345	95.101.110.185	10.10.11.249	HTTP	241	HTTP/1.1 200 OK (text/plain)	DLinkInterna_7b:2
29	3.121345	95.101.110.185	10.10.11.249	TCP	54	80 → 49859 [FIN, ACK] Seq=188 Ack=112 Win=64256 Len=0	DLinkInterna_7b:2

> Frame 1: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface \Devic...
 > Ethernet II, Src: DLink_a5:62:95 (64:29:43:a5:62:95), Dst: Intel_3a:2e:49 (68:05:ca:3a...)
 > Internet Protocol Version 4, Src: 10.10.11.249, Dst: 142.250.200.227
 > Transmission Control Protocol, Src Port: 49808, Dst Port: 80, Seq: 1, Ack: 1, Len: 1

Décrivez le mécanisme de connexion avec un diagramme.

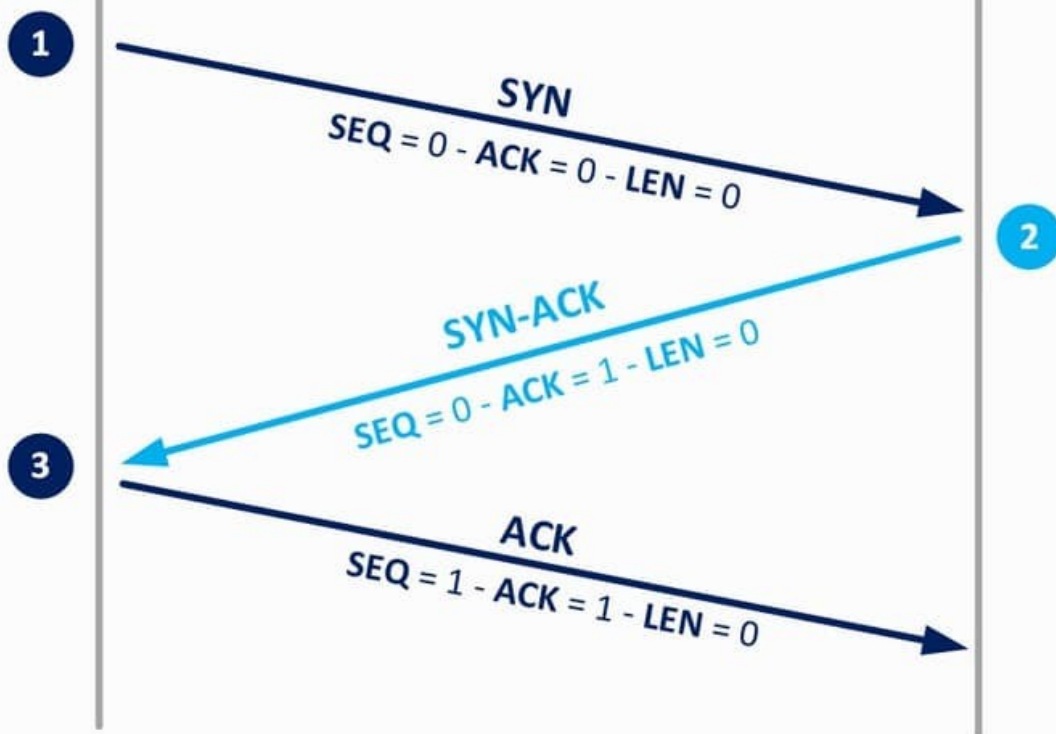
Suivi d'une connexion TCP



Poste client
192.168.100.101



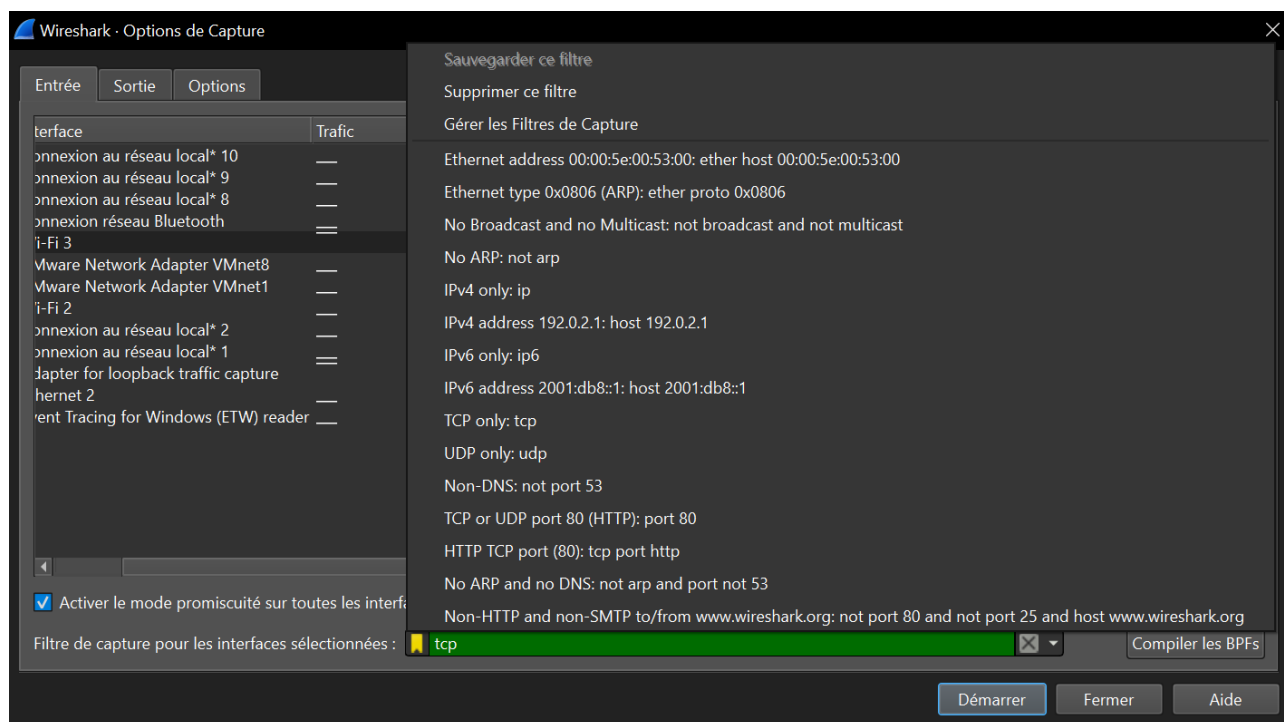
Serveur Web
192.168.100.120



Comme vous pouvez le constater, lorsqu'on écoute un réseau, beaucoup de

messages circulent. Trouver le message qui nous intéresse revient parfois à chercher une aiguille dans une botte de foin !
Il est donc fondamental d'apprendre à utiliser les filtres de Wireshark.
Documentez-vous sur le sujet et faites quelques tests pour n'afficher que les trames qui vous intéressent.

Dans l'onglet capture et dans option, on peut sélectionner plusieurs filtres prédéfinis en bas de la fenêtre en cliquant sur le logo à gauche de la barre.



On peut appliquer plusieurs filtres en même temps en séparant chaque élément par "||"

