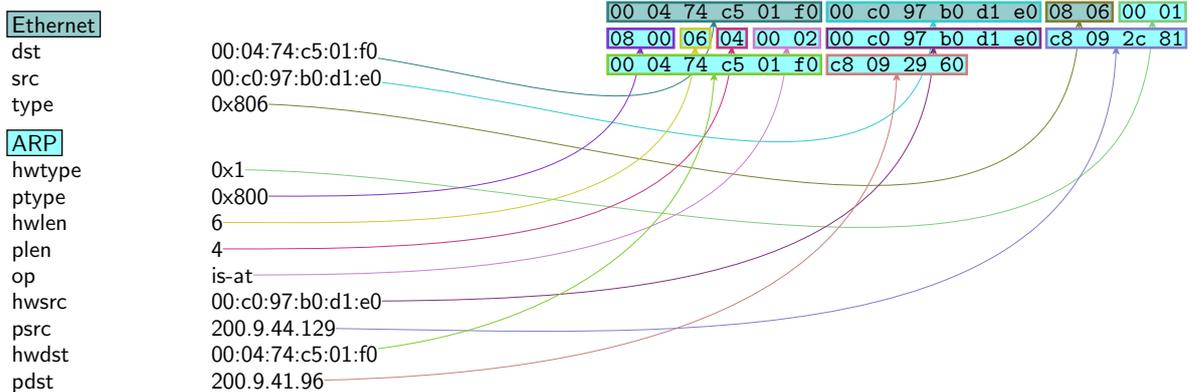


Analyse de trame

■ ■ ■ Audit réseau

1 – Première trame :



Analyse :

- * la trame contient un message ARP qui :
 - ◊ ne traverse pas les routeurs !
 - ◊ est diffusé en **broadcast** sur un réseau local au format Ethernet pour la **requête** ;
 - ◊ est diffusé en **unicast** pour la **réponse** ;
- * Ici, on a une réponse ARP avec l'opération « is-at » ;
- * On apprend que les machines d'adresse IP : 200 . 9 . 44 . 129 et 200 . 9 . 41 . 96 font donc partie du même réseau local ;
- * Si on analyse le premier octet qui diffère entre les deux adresses, c-à-d. le troisième :
 - ◊ pour 44 :

0	0	1	0	1	1	0	0
128	64	32	16	8	4	2	1

- ◊ pour 41 :

0	0	1	0	1	0	0	1
128	64	32	16	8	4	2	1

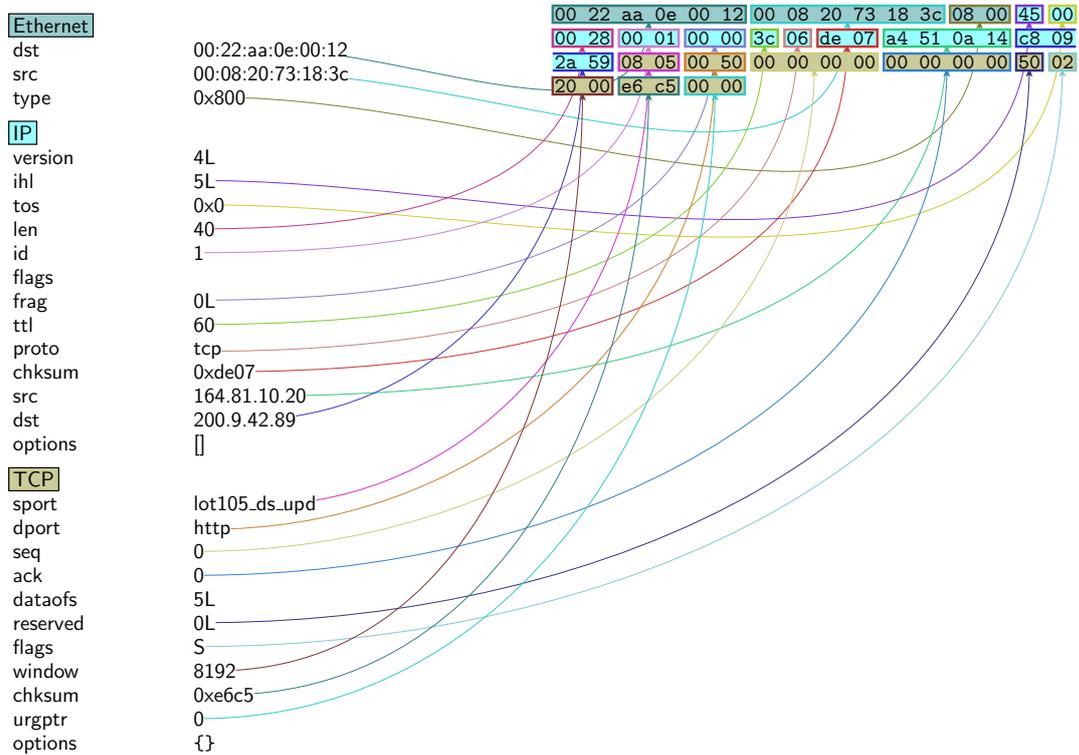
- ◊ on remarque que la partie fixe du préfixe indiquant le réseau local d'appartenance de ces deux machines est au plus :

0	0	1	0	1	0	0	0
128	64	32	16	8	4	2	1

Soit un préfixe réseau /21, qui ne respecte pas la notion de classe :

- * A, préfixe 0 : /8 ;
- * B, préfixe 10 : /16 ;
- * C, préfixe 110 : /24

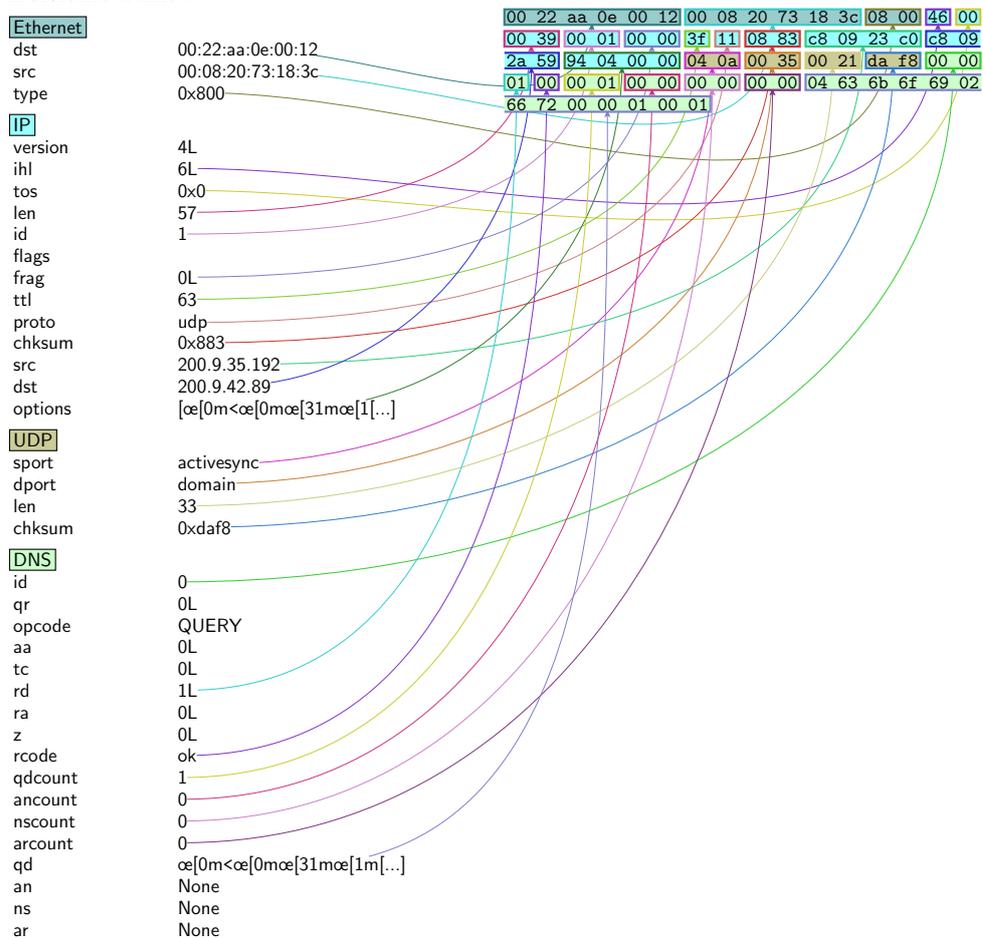
Seconde trame :



Analyse :

- * la source du datagramme est clairement à l'extérieur de notre réseau : 164.81.10.20, ce qui veut dire que cette trame provient du routeur (@MAC : 00:08:20:73:18:3c);
- * le datagramme IP contient un segment TCP de demande de connexion (drapeau SYN) vers le port 80 associé au protocole HTTP ;
- * un serveur Web serait donc hébergé sur la machine destination : 200.9.42.89 mais on a pas de confirmation : un segment SYN/ACK apporterait cette confirmation car la machine accepterait la connexion sur le port 80.
- * on apprend rien de plus sur notre réseau local (et les informations sont compatibles avec celles trouvées précédemment).

Troisième trame :



Analyse :

* on remarque que l'@MAC source est celle du routeur identifié précédemment (00 : 08 : 20 : 73 : 18 : 3c) et que l'@ IP source est 200 . 9 . 35 . 192 ;

* d'après l'analyse des @IP :

◊ pour l'@ IP source, si on décompose le troisième octet qui diffère entre les deux, c-à-d celui valant 35 :

0	0	1	0	0	0	1	1
128	64	32	16	8	4	2	1

◊ pour l'@ IP destination, on obtient pour la valeur 42 :

0	0	1	0	1	0	1	0
128	64	32	16	8	4	2	1

◊ On peut en déduire que le réseau global a été « subnetté » et que le /21 trouvé précédemment est un des sous-réseaux résultant de ce « subnettage » :
 ⇒ Le réseau global pourrait être en /20 et avoir été décomposé en deux sous réseaux /21 reliés par un routeur.

* le datagramme contient des options (taille d'en-tête de 24 > 20), il faut donc sauter ces options pour pouvoir trouver le datagramme UDP ;

* d'après l'analyse du datagramme UDP, on découvre que la machine 200 . 9 . 42 . 89 héberge un serveur DNS (port 53).

Cette information est renforcée par le contenu du datagramme UDP, c-à-d la chaîne « ckoi.fr ».

Par contre l'analyse du contenu DNS indique une « query » c-à-d une requête, donc là aussi pas de confirmation de la présence du serveur qui serait donnée par une réponse DNS.

2 – Analyses :

de la première trame :

```
###[ Ethernet ]###
  dst= 00:22:aa:01:21:31
  src= 00:d0:f1:10:12:13
  type= 0x800
###[ IP ]###
  version= 4L
  ihl= 5L
  tos= 0x0
  len= 28
  id= 567
  flags= MF
  frag= 0L
  ttl= 35
  proto= tcp
  chksum= 0xd92f
  src= 193.50.185.18
  dst= 201.27.89.21
  \options\
###[ Raw ]###
  load= '\x08\x10\x01\xbb\x00\x00\x00\x00'
```

et de la seconde :

```
###[ Ethernet ]###
  dst= 00:22:aa:01:21:31
  src= 00:d0:f1:10:12:13
  type= 0x800
###[ IP ]###
  version= 4L
  ihl= 5L
  tos= 0x0
  len= 28
  id= 567
  flags= MF
  frag= 1L
  ttl= 35
  proto= tcp
  chksum= 0xd92e
  src= 193.50.185.18
  dst= 201.27.89.21
  \options\
###[ Raw ]###
  load= '\x00\x00\x00\x00P\x02 \x00'
```

On peut observer que les datagrammes IP contenus dans chacune de ces trames ont les mêmes informations :

- * que le contenu du datagramme correspond au protocole TCP, mais que les données du datagramme sont inférieures à la taille de l'en-tête normale d'un segment TCP : 8 au lieu de 20 octets au moins pour l'en-tête ;
- * le même identifiant : 567 ;
- * le drapeau MF, indiquant la présence d'un *fragment* suivant (« *More Fragment* ») ;
- * un « *fragment offset* », de 0 pour le premier et de 1 pour le second. Chaque offset étant à multiplier par 8, cela donne pour le premier déplacement 0, et pour le second 8 par rapport au début du datagramme complet. On remarque également que la taille du datagramme est de 28, soient 20 octets d'en-tête + 8 octets de données.

Conclusion : ce sont les premier et second fragment d'un datagramme.

Si on recompose les morceaux on trouve :

```
###[ TCP ]###
  sport= 2064
  dport= 443
  seq= 0
  ack= 0
  dataofs= 5L
  reserved= 0L
  flags= S
  window= 8192
  chksum= 0xe9a1
  urgptr= 0
  options= {}
```

C'est-à-dire une connexion TCP (drapeau SYN) vers le port 443 (https).