

TD3 - Analyse de trames et routage

Avec corrections

Exercice 1.

Voici un paquet IP contenant un datagramme TCP contenant un extrait de connexion FTP. Le contenu du paquet est donné en format hexadécimal et chaque ligne représente 16 octets.

```
0x00: 45 00 00 48|5d cb 40 00|3a 06 16 2f|0b 0c 0d 0e|
0x10: 0b 0c 0d 0f|00 15 04 01|f7 90 50 b5|18 fa 80 3f|
0x20: 80 18 00 2e|47 f2 00 00|01 01 08 0a 1c 92 0d 8a
0x30: 00 3a b7 ac 53 53 48 2d 32 2e 30 2d 4f 70 65 6e
0x40: 53 53 48 5f 34 2e 33 0a
```

Chaque caractère ' | ' de la trame correspond à la fin d'une ligne dans la description de format correspondante¹, donc tous les 4 octets. Vous trouverez en annexe A les formats des en-têtes IP et TCP. Vous pouvez la dégraffer pour plus de facilités. L'en-tête IP commence donc par 45 00 et se termine par 0d 0f (20 octets en tout), et l'entête TCP commence par 00 15 et se termine par 00 00 (20 octets en tout).

1. Quelles sont les adresses IP de la source et du destinataire?

Source : 0b 0c 0d 0e -> 11.12.13.14

Destination : 0b 0c 0d 0f -> 11.12.13.15

1. On précise que ces caractères ont été rajoutés à des fins pédagogiques. Ils ne seront jamais présent sur une trame classique.

2. Quel est le numéro de port de la source ? À votre avis, la source correspond-t-elle à un client ou à un serveur ? Justifier votre réponse.

En hexadécimal, le port source est 00 15 qui se converti en 21 en décimal. 21 fait partie des ports réservés car il est compris entre 0 et 1023, en particulier il correspond au serveur FTP. À noter que certains serveurs sont tout de même sur des ports supérieurs à 1023, par exemple bittorrent utilise le port 6969 pour le tracker.

3. Même question pour le port destination.

En hexadécimal, le port destination est 04 01, c'est à dire, 1025. Il ne fait pas partie des ports réservés, donc il correspond a priori à un client.

Exercice 2.

On rappelle qu'une trame Ethernet est composé d'un en-tête de 14 octets, d'au moins 46 octets de données et de 4 octets pour le code CRC. Vous trouverez en annexe A des détails sur le contenu d'un en-tête Ethernet. On considère une capture tcpdump de trames visualisée grâce à l'outil Wireshark (voir Figure 1). La partie supérieure de la fenêtre représente toutes les trames de la capture et la partie inférieure représente le détail de la trame 6 en surbrillance.

1. Décrire précisément le rôle des trames n° 1 et 2 ?

Trame n° 1 : Dans le réseau Ethernet, la machine 192.168.0.4 effectue une requête ARP en Broadcast pour trouver l'adresse MAC de 192.168.0.1.

Trame n° 2 : Réponse ARP, la machine 192.168.0.1 envoie son adresse MAC à 192.168.0.4.

2. Quel est le rôles des trames n° 3 à 5 ? De même, quel est le rôle des trames n° 9 à 12 ?

Trame n° 3 à 5 : Ouverture d'une connexion TCP/IP de 192.168.0.4 (client) vers 192.168.0.1 (serveur sur port 80). On voit la traditionnelle poignée de main TCP/IP en trois temps : SYN, puis SYN-ACK, puis ACK.

Trame n° 9 à 12 : Fermeture de la connexion TCP/IP (flag FIN) à la demande du serveur (192.168.0.1).

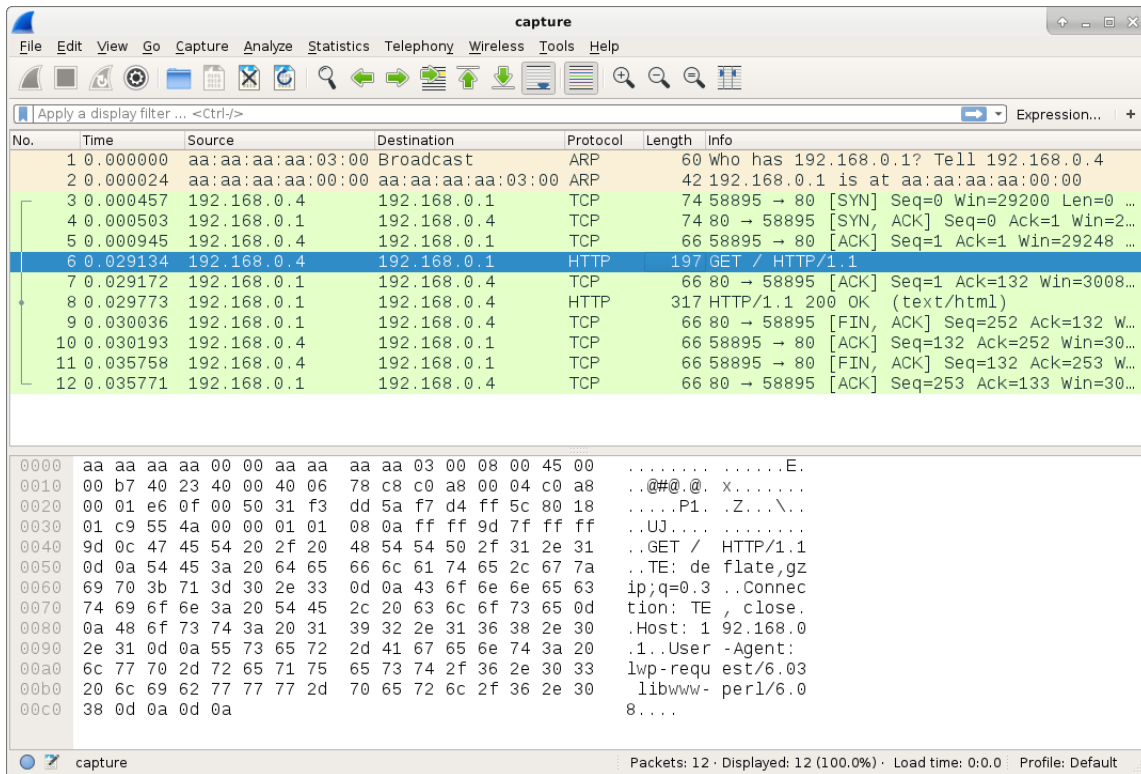


FIGURE 1 – Capture tcpdump visualisé avec Wireshark

3. A quoi correspond selon vous l'échange des trames n° 6 et 8 ?

Trame n° 6 : Il s'agit d'une requête HTTP « GET / HTTP/1.1 » du client web (192.168.0.4) vers le serveur web (192.168.0.1).

Trame n° 8 : Réponse du serveur web de la page web demandée (« / » est la page d'accueil) sous format text/html.

On se concentre maintenant sur le détail de la trame n° 6.

4. Quelle est l'adresse MAC source et destination ? Quelle est la valeur correspondant au protocole réseau (en hexa) ? Il s'agit du protocole IPv4.

Analyse du header Ethernet.

@MAC destination : aa aa aa aa 00 00

@MAC source : aa aa aa aa 03 00

EtherType : 08 00 (La trame Ethernet contient un paquet IPv4.)

-
5. Dans le paquet IP, quelle est la valeur du champs hdrl (HeaDeR Length) ? Ce nombre représente la longueur de l'en-tête du paquet IP, comptée en mots de 32 bits. En déduire la taille en octets de cet en-tête ?

IHL vaut 5 (dans le 1er octet, deuxième caractère hexa), ce qui indique une longueur de $5 \times 4 = 20$ octets pour le header IP.

6. Donner l'écriture en hexadécimal des adresses IP de la source et du destinataire sans faire de calculs.

On peut les lire sur la trame directement.

Source : c0 a8 00 04 (192.128.0.4)

Destinataire : c0 a8 00 01 (192.168.0.1)

7. Que trouve-t-on immédiatement après l'en-tête IP ? Que représente les 4 premiers octets ? Décodez ces valeurs.

On trouve le header du segment TCP. Les 4 premiers octets représentent le port source (2 octets) et le port destination du protocole TCP qui ont respectivement les valeurs 58895 (e6 0f) et 80 (00 50) en décimal.

Exercice 3 (Routage).

Voici un extrait de l'état d'une machine :

```
$ /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:e0:81:59:41:61
          inet  adr:10.0.231.3  Bcast:10.0.231.255  Masque:255.255.255.0
          adr inet6: fe80::2e0:81ff:fe59:4161/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:128132681 errors:0 dropped:30245 overruns:0 frame:0
          TX packets:102746056 errors:0 dropped:0 overruns:0 carrier:0
          Octets reçus:71419922701 (71.4 GB) Octets transmis:53179042712 (53.1 GB)
```

```
$ /sbin/route
Destination      Passerelle      Genmask          Indic Metric Ref      Use Iface
10.0.231.0       *                255.255.255.0   U      0      0        0 eth0
default          10.0.231.253    0.0.0.0         UG     100    0        0 eth0
```

```
$ arp
Address          HWtype  HWaddress        Flags Mask          Iface
10.0.231.253    ether   00:18:18:e8:0f:e0  C                   eth0
```

1. Quelle est l'adresse MAC de la machine sur laquelle les commandes précédentes ont été lancés ? Par quelle machine doit-on passer pour toute communication hors du réseau local ? Donner son adresse MAC.

L'adresse MAC de notre machine est 00:e0:81:59:41:61. Pour envoyer un message à une machine sur le réseau, on doit passer par la passerelle du réseau local. Dans notre cas, la passerelle est la machine d'IP 10.0.231.253 et son adresse MAC est lisible dans le cache arp : 00:18:18:e8:0f:e0.

2. Lorsque l'on se connecte à 10.0.231.4 , quel est le chemin des paquets ? Est-ce qu'il a été nécessaire d'utiliser le protocole ARP ? Précisez les adresses MAC des trames émises pour les paquets de notre connexion.

10.0.231.4 est compris dans le réseau directement accessible (10.0.231.0/255.255.255.0). Les paquets peuvent donc être envoyés directement de notre machine à l'autre machine. La table ARP ne contenait pas d'entrée pour 10.0.231.4, il a donc été nécessaire d'utiliser ARP pour connaître l'adresse MAC de la carte réseau de l'autre machine. Le paquet étant émis directement, l'adresse MAC source est celle de la carte réseau de notre machine, 00:e0:81:59:41:61, et l'adresse MAC destination est celle de la carte réseau de l'autre machine, telle qu'obtenue avec ARP.

-
3. Même question pour se connecter à 10.0.230.4 . Expliquez pourquoi le comportement est différent.

10.0.230.4 n'est pas compris dans le réseau directement accessible. Les paquets doivent donc être émis vers la passerelle (10.0.231.253). Il n'y a pas besoin d'utiliser ARP, car l'adresse IP de la passerelle est déjà dans la table ARP. Le paquet étant émis vers la passerelle, l'adresse MAC source est celle de la carte réseau de notre machine, 00 :e0 :81 :59 :41 :61, mais l'adresse MAC destination est celle de la passerelle, visible dans la table ARP : 00 :18 :18 :e8 :0f :e0

4. Lorsque l'on essaie de se connecter à 10.0.232.4 , on n'obtient pas de réponse. Pourtant lorsque l'on se connecte depuis la machine dont l'adresse IP est 10.0.232.3 , on obtient bien une réponse. Expliquer pourquoi cela est possible.

10.0.232.4 est sur un autre réseau que notre réseau 10.0.231.0/255.255.255.0, les paquets sont donc obligés de passer par la passerelle. Il est possible que le firewall de la passerelle refuse de laisser passer cela, pour une raison politique quelconque. Par contre, La machine dont l'adresse IP est 10.0.232.3 est très probablement dans le même réseau que 10.0.232.4, et donc son paquet ping n'a pas besoin de passer par la passerelle, et peut donc directement atteindre avec succès la destination.

5. Voici la version moderne de ces outils :

```
$ ip addr ls
```

```
1: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
  link/ether 00:e0:81:59:41:61 brd ff:ff:ff:ff:ff:ff
  inet 10.0.231.3/24 brd 10.0.231.255 scope global eth0
    valid_lft forever preferred_lft forever
  inet6 fe80::2e0:81ff:fe59:4161/64 scope link
    valid_lft forever preferred_lft forever
```

```
$ ip route ls
```

```
default via 10.0.231.253 dev eth0 onlink
10.0.231.0/24 dev eth0 proto kernel scope link src 10.0.231.3
```

```
$ ip neigh ls
10.0.231.253 dev eth0 lladdr 00:18:18:e8:0f:e0 REACHABLE
```

Repérer les mêmes informations utiles. Sous Windows, c'est encore un autre format :

```
Adresse IP. . . . . : 10.0.231.3
Masque de sous-réseau . . . . . : 255.255.255.0
Adresse IP. . . . . : fe80::2e0:81ff:fe59:4161%4
Passerelle par défaut . . . . . : 10.0.231.253
```

Et sur un OpenBSD c'est un format, sur un routeur Cisco encore un autre, etc. Heureusement que les adresse ne s'écrivent pas de la même façon !

A Annexes

Format d'en-tête Ethernet

@MAC dest.	@MAC source	prot.	data	CRC
------------	-------------	-------	------	-----

- Adresse source : 6 octets
- Adresse destination : 6 octets
- protocole : 2 octets
- données : au moins 46 octets
- CRC : 4 octets

Format d'en-tête IPv4

0	4	8	16	18	32
Ver	hdrl	TOS	length		
identification			flags	offset	
TTL		protocol	checksum		
source					
destination					
data...					

Format d'en-tête TCP

0	4	8	16	32	
source			destination		
sequence number					
acknowledgment number					
offs	res	flags	window size		
checksum			urgent pointer		
data...					