Exercices – Corrigé

Exercice 1 — Corrigé

Le champ Type vaut **0x0800**, ce qui indique que la trame Ethernet encapsule un paquet **IPv4**.

Analyse générale

- Ethernet II
- Protocole réseau : IPv4
- Protocole transport : à déterminer dans l'en-tête IP

Paramètres demandés

- Adresse IP de la machine ayant initié l'échange → Adresse IP source (champ Source IP de l'en-tête IPv4)
- Adresse MAC de la machine ayant initié l'échange → Adresse MAC source (champ SA de la trame Ethernet)
- Adresse IP de la machine ayant répondu → Adresse IP destination (champ Destination IP)
- Adresse MAC de la machine ayant répondu → Adresse MAC destination (champ DA)
- Total Length → Champ `Total Length` de l'en-tête IPv4 (longueur totale du datagramme IP)
- Protocole encapsulé → Champ `Protocol` de l'en-tête IP (ex. ICMP = 1, TCP = 6, UDP = 17)

Exercice 2 — Corrigé

Trame analysée

```
aa aa aa aa aa aa aa ab ...
```

Questions

- Que représentent les 8 octets de début ?

  Les 8 octets correspondent à :

  - 7 octets de **préambule**
  - 1 octet de **SFD** (Start Frame Delimiter)

- Donner les adresses MAC du destinataire et de l'émetteur

  Après le SFD :

  - Adresse MAC destination : `00:40:07:03:04:2b`

    - Adresse MAC source : `02:60:8c:e8:02:91`

- Donner le protocole encapsulé dans la trame

  Le champ Type vaut : `08 00` → **IPv4**

- Que représentent les 4 octets de la fin ?

  Il s'agit du champ **FCS (Frame Check Sequence)**, contenant un **CRC-32** utilisé pour la détection d'erreurs de transmission.

Exercice 3 — Corrigé

Trame 1

Ethernet

- MAC destination : `00:12:17:41:c2:c7`
- MAC source : `00:1a:73:24:44:89`
- Type : `0x0800` → IPv4

IP

- Version : 4
- Longueur d'en-tête : 20 octets (IHL = 5)
- TOS : `0x00`
- Longueur totale : `0x003c` = 60 octets
- Identifiant : `0x2730`
- DF = 0, MF = 0, Offset = 0 → non fragmenté
- TTL : `0x80` = 128
- Protocole : `0x01` → ICMP
- IP source : `192.168.1.105`
- IP destination : `192.168.1.1`

ICMP

- Type : 8
- Code : 0
- Message : **Echo Request (ping)**
- Données : chaîne ASCII `abcdefghijklmnopqrstuvwxyz...`

Trame 2

Même analyse, mais :

- Type ICMP : 0
- Code : 0
- Message : **Echo Reply**
- IP source et destination inversées

Exercice 4 — Corrigé

> Correction préalable : le protocole encapsulé est **TCP**, pas ICMP.

Trame Ethernet

- MAC destination : `00:12:17:41:c2:c7`
- MAC source : `00:1a:73:24:44:89`
- Type : `0x0800` → IPv4

Paquet IP

- Version : IPv4
- Longueur d'en-tête : 20 octets
- TOS : `0x00`
- Longueur totale : `0x01bb` = 443 octets
- Identifiant : `0xdac2`
- DF = 1, MF = 0, Offset = 0 → non fragmenté
- TTL : 60
- Protocole : `0x06` → TCP
- IP source : `213.228.0.42`
- IP destination : `62.147.81.59`

Segment TCP

- Port source : 80 (HTTP)
- Port destination : client (>1024)
- Données : contenu HTTP (réponse serveur)

Exercice 5 — Corrigé (synthèse)

Trame 1

- Ethernet II
- IPv4
- TCP
- Application : **HTTP/1.1 200 OK**

Trame 2

- Ethernet multicast
- IPv6
- ICMPv6
- Message : **Neighbor Solicitation / Advertisement**

Trame 3

- Ethernet broadcast

- Type : `0x0806`
- Protocole : **ARP Request**

Trame 4

- Ethernet II
- IPv4
- ICMP
- Message : **Echo Request (ping)**

Trame 5

- Ethernet multicast IPv6
- UDP
- Protocole applicatif : **DHCPv6**
- Message : annonce réseau (client ou serveur)
