R 2.05

2023 - 2024

# Introduction aux services réseaux

## TD N°2
## « Analyses de Trames »

***ANNE Jean-François***
***D'après le cours de M. CIREDDU R. & BAUDRU N.***

Le but de ce TD est de se familiariser avec les protocoles réseaux.

# Analyses de Trames

## 1°) Exercice 1 :

Lors d'un dialogue réseau, la trame suivante a été capturée :

```
0000 08 00 3e 26 75 2f 00 0f 1f 83 f2 26 08 00 45 00 ..>&u/.. ...&..E.
0010 00 2b 06 46 40 00 80 06 c6 0d 96 32 01 14 96 32 .+.F@... ...2...2
0020 01 01 04 5e 00 17 49 9b c3 a5 01 7c dc 05 50 18 ...^..I. ...|..P.
0030 ff fc 2e 97 00 00 ff fd 01 ........
```

Au niveau du paquet IP, aucune option et « padding » ne sont effectués.
Les adresses IP seront notées en décimale pointée et les numéros de port en base 10.

***Question :***

***Complétez le tableau ci-dessous.***

| | |
|---|---|
| **Adresse MAC source :** | **Adresse MAC destination :** |
| **Protocole de la couche 3 utilisé :** | |
| **Version Ip :** | **IHL :** |
| **Type de service :** | **Longueur totale :** |
| **Identification :** | **0 DF MF Offset :** |
| **Trame IP fragmentée ? :** | **Numéro du fragment IP :** |
| **Durée vie - TTL :** | **Checksum :** |
| **Protocole porté par le paquet IP :** | |
| **Adresse IP source :** | **Adresse IP destination :** |
| **Numéro de port source :** | **Numéro de port destination :** |

## 2°) ***Exercice 2 :***

Lors d'un dialogue réseau, la trame suivante a été capturée :

```
0000   00 12 17 41 c2 c7 00 1a 73 24 44 89 08 00 45 00
0010   00 3c 27 30 00 00 80 01 8f d6 c0 a8 01 69 c0 a8
0020   01 01 08 00 4d 56 00 01 00 05 61 62 63 64 65 66
0030   67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76
0040   77 61 62 63 64 65 66 67 68 69 ........
```

***Question :***

***Complétez le tableau ci-dessous.***

| | |
|---|---|
| **Adresse MAC source :** | **Adresse MAC destination :** |
| **Protocole de la couche 3 utilisé :** | |
| **Version Ip :** | **IHL :** |
| **Type de service :** | **Longueur totale :** |
| **Identification :** | **0 DF MF Offset :** |
| **Trame IP fragmentée ? :** | **Numéro du fragment IP :** |
| **Durée vie - TTL :** | **Checksum :** |
| **Protocole porté par le paquet IP :** | |
| **Adresse IP source :** | **Adresse IP destination :** |
| **Type et code :** | **Checksum :** |
| **Identifiant :** | **No de séquence :** |

## 3°) Exercice 3 : Analyse de protocoles

1) On considère la trace suivante, obtenue par l'analyseur de protocoles Ethereal installé sur la machine émettrice de la première trame Ethernet (les trames sont données sans préambule, ni CRC) :

### Frame Number : 1

```
0000 00 0a b7 a3 4a 00 00 01 02 6f 5e 9b 08 00 45 00
0010 00 28 00 00 40 00 40 01 82 ae 84 e3 3d 17 c2 c7
0020 49 0a 08 00 75 da 9c 7a 00 00 d4 45 a6 3a 62 2a
0030 09 00 ff ff ff ff 00 00 00 00 00 00
```

### Frame Number : 2

```
0000 00 01 02 6f 5e 9b 00 0a b7 a3 4a 00 08 00 45 00
0010 00 28 d0 92 00 00 3a 01 5a db c2 c7 49 0a 84 e3
0020 3d 17 00 00 7d da 9c 7a 00 00 d4 45 a6 3a 62 2a
0030 09 00 ff ff ff ff 00 00 00 00 00 00
```

a) Quelle est l'adresse IP (en décimal pointé) de la machine ayant initié l'échange ?

Quelle est sa classe d'adresse ?

b) Quelle est « l'adresse physique » (MAC) de la machine ayant initié l'échange ?

c) Quelle est l'adresse IP (en décimal pointé) de la machine ayant répondu ?

Quelle est sa classe d'adresse ?

d) Quelle est « l'adresse physique » de la machine ayant répondu ?

e) En supposant que la route de retour coïncide avec la route de l'aller, combien de routeurs séparent la machine source de la machine destination ?

f) Expliquez pourquoi dans les deux trames, la fin du paquet ne coïncide pas avec la fin de la trame ?

g) D'après vous, quel genre d'application, de programme ou de commande a pu générer cet échange sur le réseau ?

## 4°) Exercice 4 :

La trame Ethernet suivante a été prélevée par un programme d'écoute d'une voie Ethernet. Cette trame est éditée par lignes de 16 octets en hexadécimal.

```
0000   08 00 20 01 b4 32 08 00 20 00 61 f3 08 00 45 00
0010   00 28 0c 39 00 00 1e 06 80 77 c0 09 c8 0b c0 09
0020   c8 01 04 50 00 15 00 06 e8 02 00 80 3e 08 50 10
0030   10 00 64 be 00 00 00 00 00 00 00 00 xx xx xx xx
```

1) Entête Ethernet :
   - Quelles sont les adresses Ethernet source et destination de la trame ?
   - Où se trouve le type de protocole réseau encapsulé dans la trame Ethernet ?
   - Quelle est sa valeur ?
2) Entête IP :
   - Quelles sont les adresses IP source et destination ?
   - Y'a-t-il une fragmentation ?
   - Quelle est la valeur du champ TTL ('Time to live', durée de vie) ?
   - Comment détermine-t-on que le protocole de transport qui utilise ce datagramme IP est TCP ?
3) Entête TCP :
   - Quels sont les numéros de port source et destination de ce segment ?

## 5°) Exercice 5 : TCP, UDP

Supposons que vous vouliez transmettre le message "Hello" à une application distante sachant que :

- le protocole utilisé pour la transmission est UDP
- le protocole utilisé par UDP est IP v4
- le protocole utilisé par IP v4 est Ethernet
- le port UDP utilisé par l'application émettrice est 13
- le port UDP utilisé par l'application destinataire est 44297
- l'adresse IP de la station émettrice est 139.124.5.29
- l'adresse IP de la station destinataire est 139.124.5.58
- l'adresse Ethernet de la station émettrice est 08:00:20:75:19:7d
- l'adresse Ethernet de la station destinataire est 08:00:20:76:3e:c8

Donnez la trame Ethernet (en hexadécimal) qui sera émise par la station émettrice.

## 6°) ***Exercice 6 :***

On a représenté ci-dessous le résultat d'une capture par le logiciel WireShark de trames Ethernet (ni le préambule, ni le FCS ne sont représentés).

### ***Trame 1***

```
0000 00 12 17 41 c2 c7 00 1a 73 24 44 89 08 00 45 00 ..A.... s$D...E.
0010 00 42 da c2 40 00 3c 06 fc 9d d5 e4 00 2a 3e 93 ....@.<......*>.
0020 51 3b 00 50 04 85 87 c7 14 d5 00 12 b0 cb 50 19 Q;.P..........P.
0030 19 20 95 45 00 00 3e 20 0a 3c 74 64 20 77 69 64 . .E..> .<td wid
0040 74 86 3d 22 33 30 25 22 20 20 68 65 69 67 68 74 th="30%" height
```

### ***Trame 2***

```
0000 00 1a 73 24 44 89 00 12 17 41 c2 c7 08 00 45 00 ..s$D... .A....E.
0010 00 2c 00 00 55 56 00 01 00 05 61 62 63 64 65 66 .i..UV.. ..abcdef
0020 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmn opqrstuv
0030 77 61 62 63 64 65 66 67 68 69                   wabcdefg hi
```

Pour chacune de ces trames,

1. Entourer en rouge, les octets composant la trame Ethernet.

   Extrayez :

   - L'adresse MAC Source
   - L'adresse MAC Destination
   - Le champ type de protocole
   - Le protocole encapsulé

2. Entourer en vert les octets composant le paquet IP contenu dans la trame Ethernet

   Extrayez :

   - La version du protocole
   - La longueur de l'entête
   - La valeur du champ TOS
   - La longueur totale du datagramme IP
   - L'identifiant affecté au datagramme
   - La valeur des champs DF, MF et fragment offset.

     En déduire si datagramme est fragmenté.
   - La valeur du champ TTL
   - Le contenu du champ protocole.

     En déduire le protocole encapsulé dans le paquet IP.
   - Les adresses IP source et destination.

## 7°) ***Exercice 7 :***

Décodez la trame Ethernet suivante en vous servant des formats joints en annexe (ne donner que les champs en gras) :

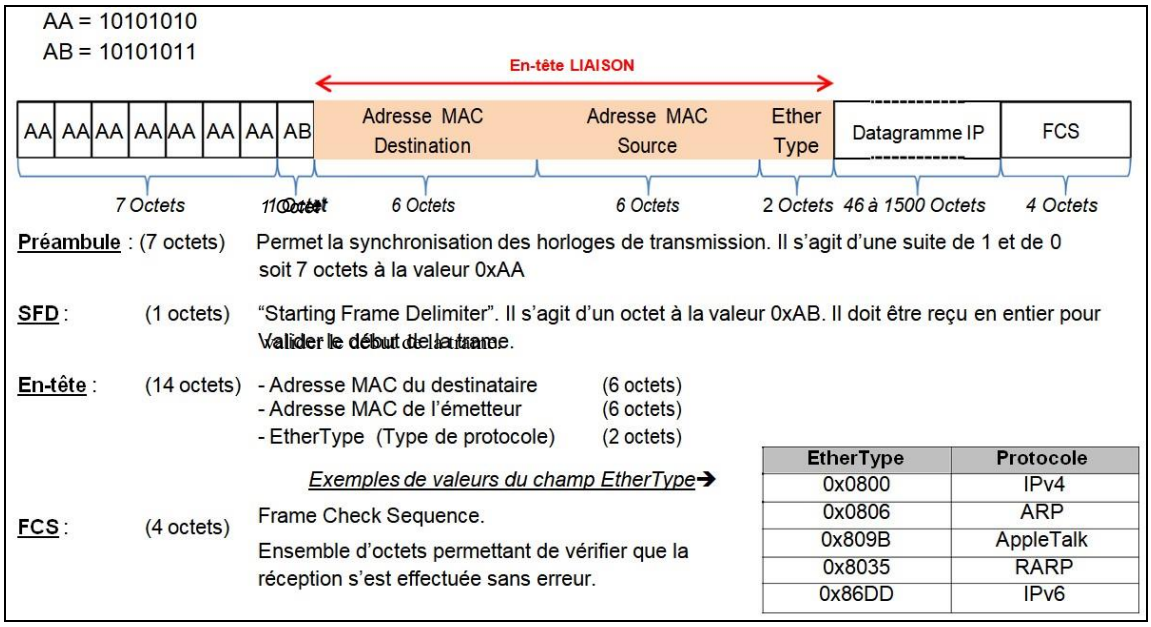
```
0000 00 04 76 f0 fb b5 00 06 5b c2 f5 9e 08 00 45 00 ..v.....[.....E.
0010 01 4f 06 cf 40 00 40 06 b1 6f c0 a8 00 17 c0 a8 .O..@.@..o......
0020 00 03 80 09 00 50 85 e6 67 33 03 6c 42 f4 80 18 .....P..g3.lB...
0030 16 d0 78 f1 00 00 01 01 08 0a 00 09 62 11 0b 5a ..x.........b..Z
0040 6a 43 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31 jCGET / HTTP/1.1
0050 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 4b 65 ..Connection: Ke
0060 65 70 2d 41 6c 69 76 65 0d 0a 55 73 65 72 2d 41 ep-Alive..User-A
0070 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e gent: Mozilla/5.
0080 30 20 28 63 6f 6d 70 61 74 69 62 6c 65 3b 20 4b 0 (compatible; K
0090 6f 6e 71 75 65 72 6f 72 2f 32 2e 32 2d 31 31 3b onqueror/2.2-11;
00a0 20 4c 69 6e 75 78 29 0d 0a 41 63 63 65 70 74 3a Linux)..Accept:
00b0 20 74 65 78 74 2f 2a 2c 20 69 6d 61 67 65 2f 6a text/*, image/j
00c0 70 65 67 2c 20 69 6d 61 67 65 2f 70 6e 67 2c 20 peg, image/png,
00d0 69 6d 61 67 65 2f 2a 2c 20 2a 2f 2a 0d 0a 41 63 image/*, */*..Ac
00e0 63 65 70 74 2d 45 6e 63 6f 64 69 6e 67 3a 20 78 cept-Encoding: x
00f0 2d 67 7a 69 70 2c 20 67 7a 69 70 2c 20 69 64 65 -gzip, gzip, ide
0100 6e 74 69 74 79 0d 0a 41 63 63 65 70 74 2d 43 68 ntity..Accept-Ch
0110 61 72 73 65 74 3a 20 41 6e 79 2c 20 75 74 66 2d arset: Any, utf-
0120 38 2c 20 2a 0d 0a 41 63 63 65 70 74 2d 4c 61 6e 8, *..Accept-Lan
0130 67 75 61 67 65 3a 20 66 72 2c 20 66 72 5f 46 52 guage: fr, fr_FR
0140 40 65 75 72 6f 2c 20 65 6e 0d 0a 48 6f 73 74 3a @euro, en..Host:
0150 20 73 65 72 76 43 33 30 39 0d 0a 0d 0a servC309....
```

Trame Ethernet Paquet IP Segment TCP

3. À votre avis, quel est le protocole transporté à l'intérieur du segment TCP et quelle est l'application qui l'utilise ? Sous quelle forme se présentent les champs de ce protocole ?

# Définitions

## I. STRUCTURE DE LA TRAME ETHERNET II




**Préambule** : (7 octets) Permet la synchronisation des horloges de transmission. Il s'agit d'une suite de 1 et de 0 soit 7 octets à la valeur 0xAA

**SFD** : (1 octets) "Starting Frame Delimiter". Il s'agit d'un octet à la valeur 0xAB. Il doit être reçu en entier pour valider le début de la trame.

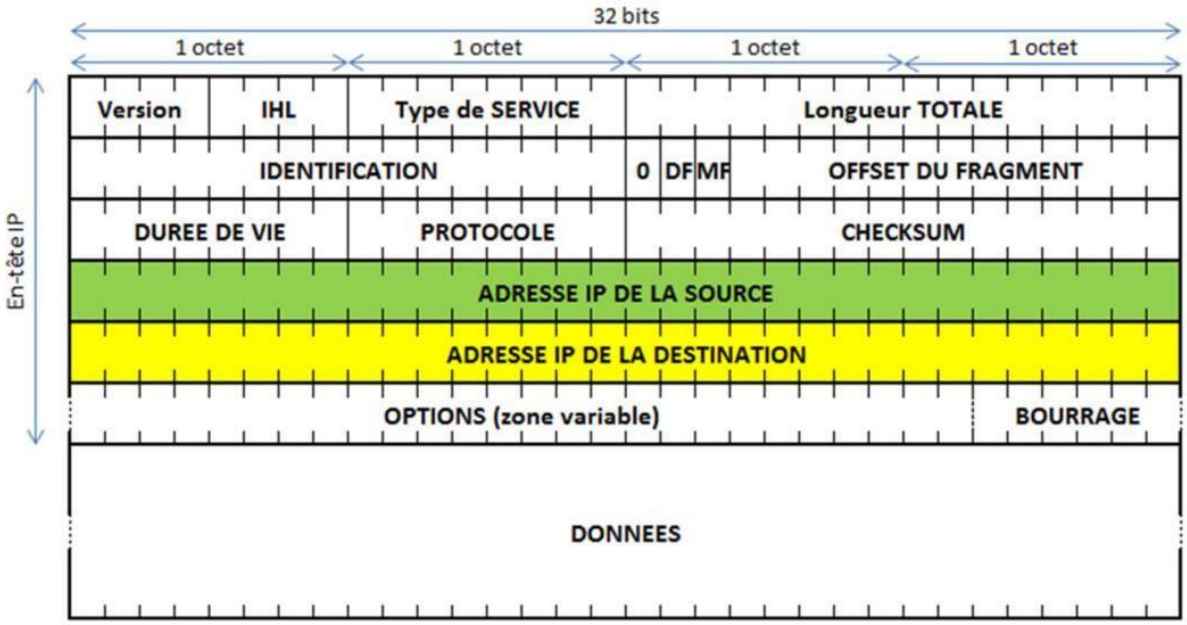
**En-tête** : (14 octets)
- Adresse MAC du destinataire (6 octets)
- Adresse MAC de l'émetteur (6 octets)
- EtherType (Type de protocole) (2 octets)

*Exemples de valeurs du champ EtherType* ➔

| EtherType | Protocole |
|---|---|
| 0x0800 | IPv4 |
| 0x0806 | ARP |
| 0x809B | AppleTalk |
| 0x8035 | RARP |
| 0x86DD | IPv6 |

**FCS** : (4 octets) Frame Check Sequence.
Ensemble d'octets permettant de vérifier que la réception s'est effectuée sans erreur.

# II. STRUCTURE D'UN PAQUET (DATAGRAMME) IP




**Version** : (4 bits) il indique le numéro de version du protocole IP utilisé (généralement 4).

**IHL** : (4 bits) Internet Header Lenght (Longueur d'entête). Spécifie la longueur de l'en-tête du Datagramme en nombre de mots de 32 bits. Ce champ ne peut prendre une valeur inférieure à 5.

**Type de service** : (8 bits) Donne une indication sur la qualité de « service » souhaitée pour l'acheminement des données.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Priorité | | | D | T | R | C | x |

| *Bits 0- 2* | **Priorité** | 010➔Immédiate 001➔Normale 000➔Basse | |
|---|---|---|---|
| *Bit 3* | **D** | 0 = Retard standard | 1 = Retard faible |
| *Bit 4* | **T** | 0 = Débit standard | 1 = Haut débit |
| *Bit 5* | **R** | 0 = Taux d'erreur standard | 1 = Taux d'erreur faible |
| *Bit 6* | **C** | 0 = Coût standard | 1 = Coût faible |
| *Bit 7* | **x** | Réservé | |

**Longueur totale** : (16 bits) Longueur du datagramme entier y compris en-tête et données mesurée en octets.

**Identification** : (16 bits) Valeur assignée par l'émetteur pour identifier les fragments d'un même datagramme.

**Flags** : (3 bits) Commutateurs de contrôle :
- Bit 0 Réservé, doit être laissé à 0
- Bit 1 (DF - Don't fragment) 0= Fragmenté 1= Non fragmenté
- Bit 2 (MF – More Fragment) 0= Dernier fragment 1= Fragment

**OFFSET** : (13 bits) Décalage du premier octet du fragment par rapport au datagramme complet non fragmenté. Cette position est mesurée en blocs de 8 octets (64 bits).

**Durée de vie** : (8 bits) Temps en secondes pendant lequel le datagramme doit rester dans le réseau. Si ce champ vaut 0, le datagramme doit être détruit. Ce temps diminue à chaque passage du datagramme d'une machine à l'autre.

**Protocole** : (8 bits) Protocole porté par le datagramme (au-dessus de la couche IP)

| Valeur | Protocole |
|---|---|
| 1 | ICMP |
| 6 | TCP |
| 17 | UDP |
| Etc | etc |

**Checksum** : (16 bits) (Somme de contrôle) C'est une valeur qui permet de déceler une éventuelle erreur de transmission avec une très grande probabilité.

**IP Source** : (32 bits) Adresse IP de l'émetteur.

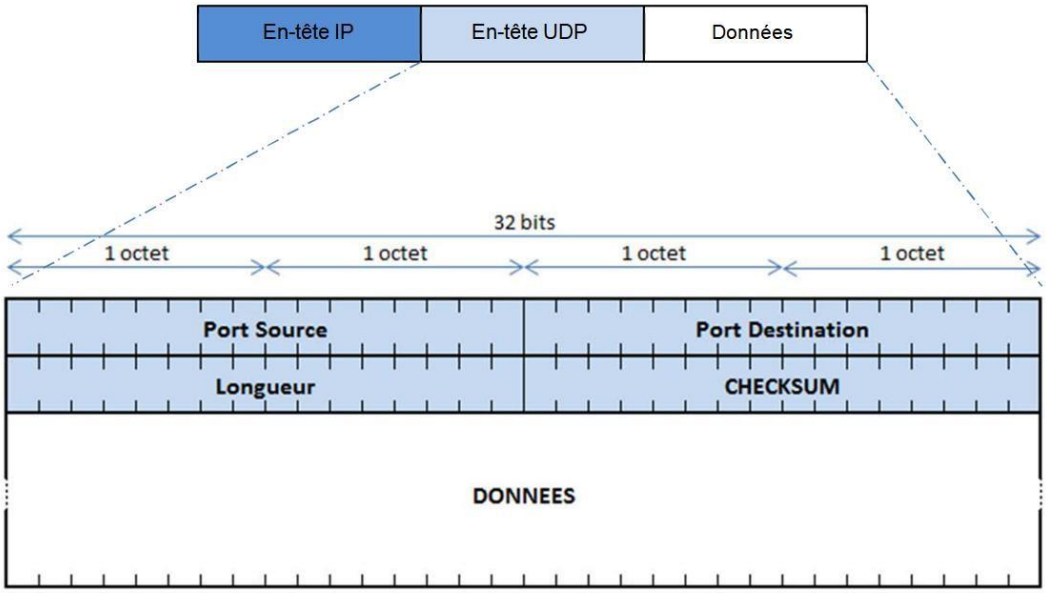
**IP Destination** : (32 bits) Adresse IP du destinataire.

**Options** : (Variable) Le champ est de longueur variable.Un datagramme peut comporter 0 ou plusieurs options.

**Bourrage** : (Variable) Le champ Bourrage n'existe que pour assurer à l'en-tête une taille totale multiple de 4 octets. Le bourrage se fait par des octets à 0.

## III. STRUCTURE D'UN SEGMENT UDP

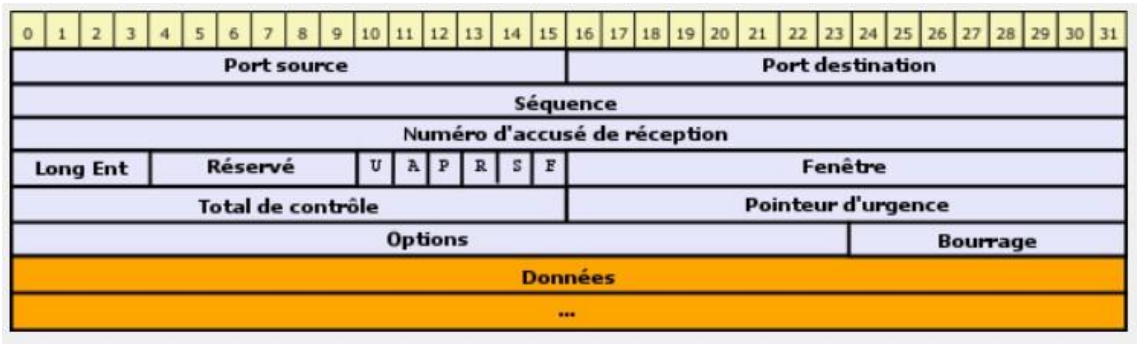
**UDP** est encapsulé dans un paquet IP. Il comporte un en-tête suivi des données proprement dites à transporter.
L'en-tête d'un datagramme UDP est plus simple que celui de TCP :




Il contient les quatre champs suivants :

**Port Source :** indique depuis quel port le paquet a été envoyé.
**Port Destination :** indique à quel port le paquet doit être envoyé.
**Longueur :** indique la longueur totale (exprimée en octets, en-tête et données). La longueur minimale est donc de 8 octets (taille de l'en-tête).
**Somme de contrôle (CHEKSUM) :** celle-ci permet de s'assurer de l'intégrité du paquet reçu quand elle est différente de zéro. Elle est calculée sur l'ensemble de l'en-tête UDP et des données, mais aussi sur un pseudo en-tête (extrait de l'en-tête IP).

# IV. STRUCTURE D'UN SEGMENT TCP




**Port source :** c'est le port utilisé pour les données à émettre. C'est un des éléments du quadruplet qui identifie une connexion.
**Port destination :** c'est le port où les données sont envoyées. Il doit être connu pour identifier l'application à laquelle les données sont envoyées. C'est le deuxième élément du quadruplet identificateur.
**Numéro de séquence :** il donne la position du segment dans le flux de l'émetteur. Deux cas sont à considérer :

- le bit SYN est positionné à 1, alors le numéro de séquence a pour valeur *ISN* [*Initial Sequence Number*] + 1.
- le bit SYN est positionné à 0, alors le numéro de séquence a pour valeur le numéro du premier octet de données relativement au début de la transmission.

**Numéro d'accusé de réception :** il indique le numéro du prochain octet attendu par le récepteur.
**Longueur en-tête :** il indique la longueur de l'en-tête d'un segment TCP et est exprimé comme un multiple de 32 bits. Ce champ est rendu indispensable dans la mesure où la longueur du champ option est variable (selon les options choisies).
**Réservé :** comme son nom l'indique, il est réservé à un usage futur. Il est donc positionné à zéro.
**Bits de code TCP :**

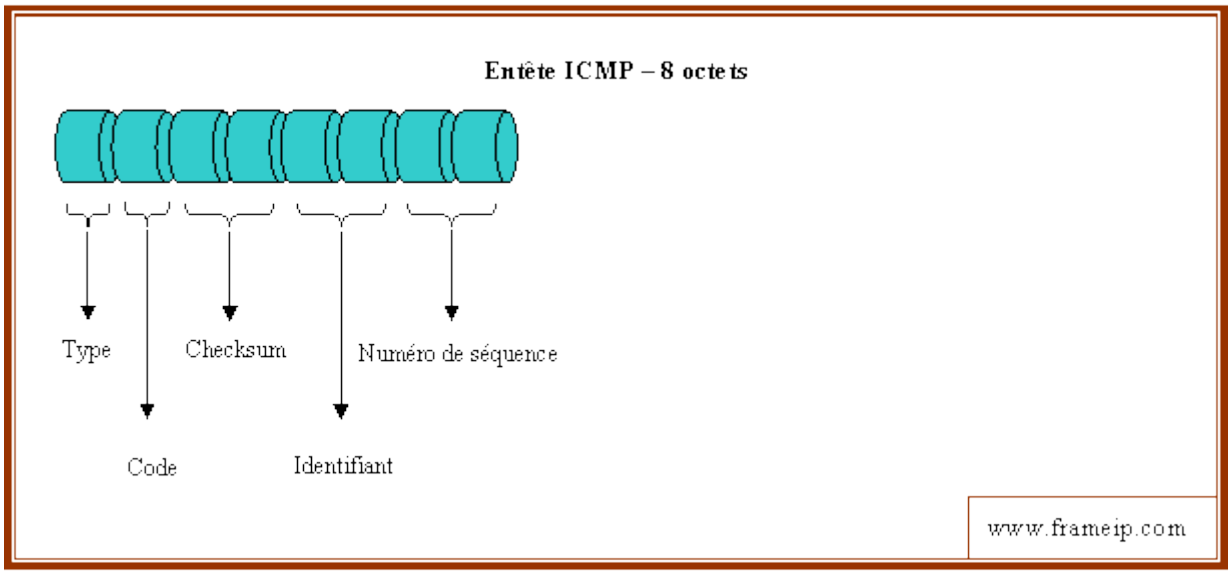
| | |
|---|---|
| bit *URG* | C'est le pointeur de données urgentes s'il est positionné à 1.<br>Indique que les données doivent être délivrées immédiatement (notification d'évènement en temps réel). |
| bit *ACK* | Le paquet est un accusé de réception, s'il est positionné à 1.<br>Le ACK flag indique que l'ASN [*Acknowledgement Sequence Number*] de l'en-tête TCP est valide et contient le prochain octet de données attendu. |
| bit *PSH* | Le segment requiert un *push*, s'il est positionné à 1.<br>Ce flag indique au recépteur que les données doivent être remise immédiatement à l'application, sans bufférisation. Utilisé dans les sessions interactive comme OpenSSH. |
| bit *RST* | Réinitialiser la connexion, s'il est positionné à 1.<br>Ce flag positionné par une des extrêmités indique une condition d'erreur non récupérable. Dans ce cas, les deux extrêmités termine la connexion, libère les ressources allouées à la connexion et détruise tous les paquets subséquents en transit. |
| bit *SYN* | Demande de synchronisation des numéros de séquence, s'il est positionné à 1.<br>Ce flag est armé dans le premier paquet envoyé par le client et le serveur. Chaque octet de données est séquentiellement numéroté (l'ISN [*Initial Sequence Number*] doit être choisi aléatoirement (éviter les prédictions), les échanges subséquent se font en incrémentant cet ISN. |
| bit *FIN* | Indique la fin d'une connexion, s'il est positionné à 1.<br>Indique que la transmission est terminé (complète), envoit d'un message SYN+FIN, attente de l'acquittement de confirmation ; message FIN puis message SYN+FIN, envoit d'un message ACK final. |

**Fenêtre :** indique le nombre d'octet que le récepteur peut admettre (à partir de la position contenue dans l'accusé de réception) sans qu'un accusé de réception soit nécessaire.
**Total de contrôle :** ce champ permet de vérifier l'intégrité de l'en-tête TCP et des données. C'est le complément à 1 (sur 16 bits) de la somme des compléments à 1 des octets de l'en-tête et des données (par mots de 32 bits). À noter que le champ de 16 bits le représentant est positionné à 0 lors du calcul.
**Pointeur d'urgence :** ce champ est utilisé si le bit *URG* est positionné (à 1) indique dans la fenêtre la position où les données urgentes s'arrêtent.

**Options :** ce champ contient les différentes options TCP. Par exemple, le *MSS* ([*Maximum Segment Size*], taille maximale des segments), le *window scale option*, le *timestamp option* ...
**Bourrage (taille variable) :** permet de parvenir à un en-tête d'une taille multiple de 32 bits. Il complète par des 0 la fin du champ *options*.
**Données (taille variable) :** il s'agit des données à transmettre.

# V. STRUCTURE D'UN SEGMENT ICMP

Voici la structure de l'entête ICMP basé sur 8 octets.




Les deux champs Identifiant et Numéro de séquence ne sont présent que dans le cas d'un paquet de type Ping sinon les champs restent présents mais en tant que bourrage et donc non utilisés.

## A. Définition des différents champs

### 1. Type et Code

Les champs Type et Code sont codés respectivement sur 8 bits ce qui donne un totale de 2 octets. Ils représentent la définition de message d'erreur contenu. Voici la liste des principales combinaisons entre les champs Type et Code :

- type=00 et code=00 : Réponse à une demande d'écho
- type=03 et code=00 : Réseau inaccessible
- type=03 et code=01 : Hôte inaccessible
- type=03 et code=02 : Protocole inaccessible
- type=03 et code=03 : Port inaccessible
- type=03 et code=04 : Fragmentation nécessaire mais interdite
- type=03 et code=05 : Echec de routage par la source
- type=03 et code=06 : Réseau de destination inconnu
- type=03 et code=07 : Hôte de destination inconnue
- type=03 et code=08 : Machine source isolée
- type=03 et code=09 : Réseau de destination interdit administrativement
- type=03 et code=10 : Hôte de destination interdite administrativement

- type=03 et code=11 : Réseau inaccessible pour ce type de service
- type=03 et code=12 : Hôte inaccessible pour ce type de service
- type=03 et code=13 : Communication interdite par un filtre
- type=03 et code=14 : Host Precedence Violation
- type=03 et code=15 : Precedence cutoff in efect
- type=04 et code=00 : Volume de donnée trop importante
- type=05 et code=00 : Redirection pour un hôte
- type=05 et code=01 : Redirection pour un hôte et pour un service donné
- type=05 et code=02 : Redirection pour un réseau
- type=05 et code=03 : Redirection pour un réseau et pour un service donné
- type=08 et code=00 : Demande d'écho
- type=09 et code=00 : Avertissement routeur
- type=10 et code=00 : Sollicitation routeur
- type=11 et code=00 : Durée de vie écoulée avant d'arrivée à destination
- type=11 et code=01 : Temps limite de réassemblage du fragment dépassé
- type=12 et code=00 : Entête IP invalide
- type=12 et code=01 : Manque d'une option obligatoire
- type=12 et code=02 : Mauvaise longueur
- type=13 et code=00 : Requête pour un marqueur temporel
- type=14 et code=00 : Réponse pour un marqueur temporel
- type=15 et code=00 : Demande d'adresse réseau
- type=16 et code=00 : Réponse d'adresse réseau
- type=17 et code=00 : Demande de masque de sous réseau
- type=18 et code=00 : Réponse de masque de sous réseau

### a) ***Type=0,8 – Le Ping***

Le principe du Ping étant, à la base, de valider la présence d'un Hote IP. Pour cela, l'application Ping utilisera la séquence 8-0 afin d'émettre une demande d'écho. Les données reçues dans un message d'écho doivent être réémises dans la réponse. Ainsi, si le message de retour correspond à l'émission, on en déduit que l'Hote est présent. De plus, on peut en déduire d'autres services, tel que le temps de réponse, la taille paquet maximum la durée de vie et etc.

L'identificateur et le numéro de séquence peuvent être utilisés par l'émetteur du message d'écho afin d'associer facilement l'écho et sa réponse. Par exemple, l'identificateur peut être utilisé comme l'est un port pour TCP ou UDP, identifiant ainsi une session. Et le numéro de séquence peut être incrémenté pour chaque message d'écho envoyé. L'hôte de destination respectera ces deux valeurs pour le retour.

### b) Type=3 – Destination non valide

Ce type de message est émis dans le cas où un routeur ou un hôte ne puisse pas router un paquet.

### c) Type=4 – Volume de donnée trop importante

Un routeur ou hôte peut être amené à détruire un Datagramme s'il manque de mémoire pour bufferiser. Dans ce cas, le routeur émettra un message à destination de la source du Datagramme détruit, un paquet ICMP de type 4.

Cela peut se produire dans un second cas. Quand le Datagramme arrive trop rapidement pour qu'il puisse être traité le message ICMP peut donc constituer une demande de diminution de débit de transfert.

### d) Type=5 – Redirection

Ces types de messages sont émis afin d'indiquer que le chemin emprunté n'est pas le bon pour la destination demandée. La réception de ce type de message d'erreur peut être interprétée par la modification de la table de routage locale. C'est plus communément appelé « ICMP Redirect ».

### e) Type=9,10 – découverte de routeur

Au démarrage d'une station, plutôt que de configurer manuellement les routes statiques, surtout si elles sont susceptibles de changer et que le nombre de stations est grands, il peut être intéressant de faire de la découverte automatique de routeurs. Pour cela, il y a deux possibilités. La première est l'envoi régulier de messages ICMP de type 9 « Avertissement routeur » d'annonces de routes par les routeurs. La seconde possibilité est qu'une station sollicite les routeurs avec un message de type 10 « Sollicitation routeur ».

La découverte de routeur n'est pas un protocole de routage, son objectif est bien moins ambitieux, juste obtenir une route par défaut.

Vous trouverez tous les détails du « découverte de routeur » dans la RFC 1256.

### f) Type=11 – Temps excédé

Lorsqu'un routeur traitant un Datagramme est amené à mettre à jour le champ Durée de Vie de l'entête IP et que ce champ atteint une valeur zéro, le Datagramme sera détruit. Le routeur peut alors envoyer un message d'erreur « Time To Live expiré ». Ce message ICMP peux être émis aussi dans le cas où le temps de réassemblage expire et le Datagramme ne peux donc être reconstitué à temps.

### g) Type=12 – Erreur d'entête

Si un routeur rencontre un problème avec un paramètre d'entête IP l'empêchant de finir son traitement, le datagramme sera alors détruit. Un message d'erreur de type 12 sera donc alors émis.

### h) Type=13,14 – Marqueur temporel

Le but de ce type de message est de s'échanger des données temporelles pour, par exemple, synchroniser les horloges.

### i) Type=15,16,17,18 – Demande d'information

Ce type de message est envoyé vers une adresse constituée du numéro de réseau dans le champ source de l'entête IP et un champ destinataire à 0. La pile IP qui répondra pourra alors envoyer une réponse avec les adresses entièrement renseignées.

## 2. Checksum

Le champ Checksum est codé sur 16 bits et représente la validité du paquet de la couche 3 ICMP. Pour pouvoir calculer le Checksum, il faut positionner le champ du checksum a 0. Ce calcul est strictement le même que celui du protocole IGMP.

### 3. Identifiant

Le champ identifiant est codé sur 16 bits et définit l'identifiant de l'émetteur. Pour cela, il est conseillé d'assigner le numéro du processus assigné (PID) à l'application lors de l'exécution. Cela permet de le rendre unique inter application. Cela ressemble beaucoup aux numéros de port pour les protocoles TCP et UDP.

### 4. Numéro de séquence

Le champ Séquence est codé sur 16 bits et permet au récepteur, d'identifier s'il manque un paquet. Le plus classique étant une incrémentation linéaire de 1. Ainsi, si le récepteur reçoit la séquence 1 puis 3, il peut en déterminer une perte d'un paquet. Néanmoins, ce n'est pas normalisé, donc personne n'à la garantie que l'émetteur utilisera cette méthode. Cela peut aussi permettre à l'émetteur d'envoyer multiples paquets et de pouvoir distinguer les retours.

# Webographie


- http://robert.cireddu.free.fr/SNIR/TD%20sur%20la%20lecture%20et%20le%20decodage%20de%20trame.pdf
- https://www.frameip.com/entete-ethernet/#4-8211structure-de-lrsquoentete-ethernet
- https://inetdoc.net/articles/ethernet/ethernet.trame.erreur.html
- https://dokumen.tips/download/link/td-4-pro-toc-oles-tcpudp-avec-correction
- http://www.linux-france.org/prj/edu/archinet/systeme/ch06s03.html
- http://cisco.teckn0.com/ccna2_final/v3/Cisco%20Networking%20Academy.htm
- http://www.mysti2d.net/polynesie2/ETT/C044/31/SerruresBioIP/index.html?Cours4.html