

 <h2 style="text-align: center;">Annexe 3 - La trame Ethernet</h2>	
<b>Sommaire :</b>	
I - Format de la trame Ethernet.....	1
II - Format du datagramme IP (Internet Protocol).....	2
III - Format du datagramme UDP (User Datagram Protocol).....	4
IV - Format du segment TCP (Transfert Control Protocol).....	4
V - Protocole ICMP (Internet Control Message Protocol).....	6
VI - Protocole ARP (Adress Resolution Protocol).....	7

### I - Format de la trame Ethernet

Le **format** d'une **trame Ethernet** est le suivant :

Préambule	Adresse Physique Destinataire	Adresse Physique Source	Type	Données	CRC
8 octets	6 octets	6 octets	2 octets	46 à 1500 octets	4 octets

**Préambule** (8 octets) :

Il est composé de 64 bits de synchronisation, alternance de **1** et **0** avec les deux derniers bits à 1 (**SFD**).

**Adresse Physique destination** (6 octets) :

Ce champ contient l'adresse physique (Ethernet) de la station devant recevoir la trame. Les trois premiers octets de cette adresse sont imposés par l'IEEE aux fabricants de contrôleurs, ce qui garantit son unicité. Il y a diffusion si tous les bits sont à 1.

**Adresse Physique source** (6 octets) :

Ce champ contient l'adresse physique (Ethernet) de la station ayant émis la trame.

**Type** (2 octets) :

Ce champ identifie le protocole de niveau supérieur associé au paquet :

- **0x0800** = Protocole **IP**,
- **0x0806** = Protocole **ARP**,
- **0x8035** = Protocole **RARP**.

**Données** (46 à 1500 octets) :

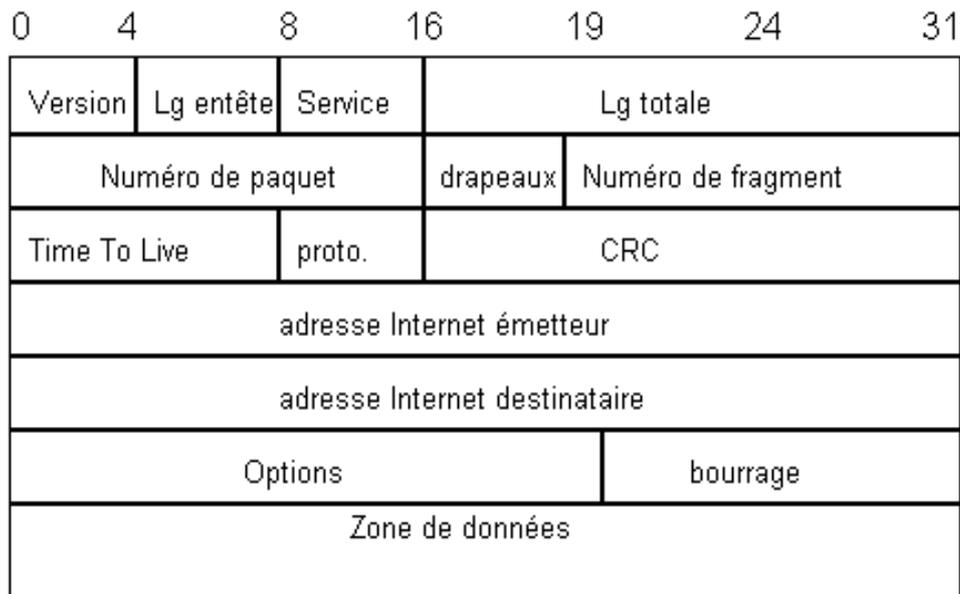
Ce champ contient les informations à transporter. La trame **ARP** ou l'entête **IP** (qui comporte les adresses Internet sources et destinataires) suivi des données propres aux protocoles de niveaux supérieurs (par exemple les données **TCP**).

**CRC** (4 octets) :

Contrôle de Redondance Cyclique (**Cyclic Redundancy Check**). C'est une somme de contrôle portant sur tout ce qui précède sauf le préambule.

## II - Format du datagramme IP (Internet Protocol)

Un **datagramme IP** a la structure suivante :



**Version** (4 bits) : Le champ Version renseigne sur le format de l'en-tête Internet. Ce document décrit le format de la version **4** du protocole.

**Longueur d'En-Tête** (4 bits) : Le champ Longueur d'En-Tête (LET) code la longueur de l'en-tête Internet, l'unité étant le mots de 32 bits, et de ce fait, marque le début des données. Notez que ce champ ne peut prendre une valeur en dessous de 5 pour être valide.

**Type de Service** (8 bits) : Le "Type de Service" sert à préciser le traitement effectué sur le datagramme pendant sa transmission à travers Internet. Principalement, le choix offert est une négociation entre les trois contraintes suivantes : faible retard, faible taux d'erreur, et haut débit.

<b>Bits 0-2 :</b>	Priorité.	
<b>Bit 3 : D</b>	0 = Retard standard,	1 = Retard faible.
<b>Bits 4 : T</b>	0 = Débit standard,	1 = Haut débit.
<b>Bits 5 : R</b>	0 = Taux d'erreur standard	1 = Taux d'erreur faible.
<b>Bits 6-7 :</b>	Réservé.	

### Bits 0-2 : Priorité :

- 111 - Network Control
- 110 - Internetwork Control
- 101 - CRITIC/ECP
- 100 - Flash Override
- 011 - Flash
- 010 - Immediate
- 001 - Priority
- 000 - Routine

**Longueur Totale** (16 bits) : Le champ "Longueur Totale" est la longueur du datagramme entier y compris en-tête et données, mesurée en octets. Ce champ ne permet de coder qu'une longueur de datagramme d'au plus 65535 octets.

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

**Flags** (3 bits) : Divers commutateurs de contrôle.

Bit 0: réservé, doit être laissé à zéro

Bit 1: (AF) 0 = Fragmentation possible, 1 = Non fractionnable.

Bit 2: (DF) 0 = Dernier fragment, 1 = Fragment intermédiaire.

**Fragment Offset (Position relative)** (13 bits) : Ce champ indique le décalage du premier octet du fragment par rapport au datagramme complet. Cette position relative est mesurée en blocs de 8 octets (64 bits). Le décalage du premier fragment vaut zéro.

**Durée de vie ou Time To Live (TTL)** (8 bits) : Ce champ permet de limiter le temps pendant lequel un datagramme reste dans le réseau. Si ce champ prend la valeur zéro, le datagramme doit être détruit. Ce champ est modifié pendant le traitement de l'en-tête Internet. La durée de vie est mesurée en secondes.

**Protocole** (8 bits) : Ce champ indique quel protocole de niveau supérieur est utilisé dans la section données du datagramme IP.

- **0x01** = Protocole **ICMP**,
- **0x02** = Protocole **IGMP**,
- **0x06** = Protocole **TCP**,
- **0x11** = Protocole **UDP**.

**Checksum d'en-tête** (16 bits) : Un Checksum calculé sur l'en-tête uniquement. Comme certains champs de l'en-tête sont modifiés (ex., durée de vie) pendant leur transit à travers le réseau, ce Checksum doit être recalculé et vérifié en chaque point du réseau où l'en-tête est réinterprétée.

L'algorithme utilisé pour le Checksum est le suivant : On calcule le complément à un sur 16 bits de la somme des compléments à un de tous les octets de l'en-tête pris par paires (mots de 16 bits).

**Adresse source** (32 bits) : L'adresse IP de la source.

**Adresse destination** (32 bits) : L'adresse IP du destinataire.

**Options** (variable) : Les datagrammes peuvent contenir des options. Celles-ci doivent être implémentées par tous les modules IP (hôtes et routeurs). Le caractère "optionnel" concerne leur transmission, et non leur implémentation.

**Bourrage** (variable) : Le champ de 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 zéro.

### III - Format du datagramme UDP (User Datagram Protocol)

Un datagramme **UDP** a la structure suivante :

<b>Port Source (16 bits)</b>	<b>Port Destination (16 bits)</b>
<b>Longueur (16 bits)</b>	<b>Somme de contrôle (16 bits)</b>
<b>Données (longueur variable)</b>	

Voici le détail des **quatre champs** qui forment l'en-tête du datagramme :

**Port Source** (16 bits) : Port relatif à l'application en cours sur la machine source ;

**Port Destination** (16 bits) : Port relatif à l'application en cours sur la machine de destination ;

**Longueur** (16 bits) : Ce champ précise la longueur en octets, du message, en-tête compris ;

**Somme de contrôle** (16 bits) : La somme de contrôle est réalisée en faisant la somme des champs de l'en-tête et des données.

Les principaux ports UDP :

Port	Protocole	Port	Protocole
53 67, 68 69	DNS BOOTP et DHCP TFTP	137 161, 162 520	NetBios SNMP RIP

### IV - Format du segment TCP (Transfert Control Protocol)

Un segment **TCP** a la structure suivante :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Port Source																Port destination															
Numéro d'ordre																															
Numéro d'accusé de réception																															
Décalage données		réservée		URG	ACK	PSH	RST	SYN	FIN	Fenêtre																					
Somme de contrôle																Pointeur d'urgence															
Options																						Remplissage									
Données																															

**Port Source** (16 bits) : Port relatif à l'application en cours sur la machine source;

**Port Destination** (16 bits) : Port relatif à l'application en cours sur la machine de destination;

**Numéro d'ordre** (32 bits) : Lorsque le drapeau SYN est à 0, le numéro d'ordre est celui du premier mot du segment en cours. Lorsque SYN est à 1, le numéro d'ordre est égal au numéro d'ordre initial utilisé pour synchroniser les numéros de séquence (ISN);

**Numéro d'accusé de réception** (32 bits) : Le numéro d'accusé de réception également appelé numéro d'acquiescement correspond au numéro (d'ordre) du prochain segment attendu, et non le numéro du dernier segment reçu;

**Décalage des données** (4 bits) : il permet de repérer le début des données dans le paquet. Le décalage est ici essentiel car le champ d'options est de taille variable

**Drapeaux ou Indicateurs (flags)** (6 x 1 bit) :

- **URG** : si ce drapeau est à 1 le paquet doit être traité de façon urgente;
- **ACK** : si ce drapeau est à 1 le paquet est un accusé de réception;
- **PSH** (PUSH) : si ce drapeau est à 1, le paquet fonctionne suivant la méthode PUSH;
- **RST** : si ce drapeau est à 1, la connexion est réinitialisée;
- **SYN** : Le Flag TCP SYN indique une demande d'établissement de connexion;
- **FIN** : si ce drapeau est à 1 la connexion s'interrompt;

**Fenêtre** (16 bits) : Champ permettant de connaître le nombre d'octets que le récepteur souhaite recevoir sans accusé de réception;

**Somme de contrôle** (Checksum ou CRC) : La somme de contrôle est réalisée en faisant la somme des champs de données de l'en-tête;

**Pointeur d'urgence** (16 bits) : Indique le numéro d'ordre à partir duquel l'information devient urgente;

**Options** (Taille variable) : Des options diverses;

**Remplissage**: On remplit avec des **0** pour avoir une longueur multiple de 32 bits.

Les principaux ports TCP :

Port	Protocole	Port	Protocole
20, 21	FTP	110	POP3
22	SSH	137-139	NetBios
23	Telnet	143	IMAP
25	SMTP	389	LDAP
80	HTTP		

## V - Protocole ICMP (Internet Control Message Protocol)

Le protocole **ICMP** (*Internet Control Message Protocol*) est un protocole qui permet de gérer les informations relatives aux **erreurs** des machines connectées. Il permet non pas de corriger les erreurs mais de faire part de ces erreurs aux protocoles des couches voisines.

Voici à quoi ressemble un **message ICMP** encapsulé dans un datagramme **IP** :

Type (8 bits)	Code (8 bits)	Checksum (16 bits)	Message (taille variable)
------------------	------------------	-----------------------	------------------------------

Type du message :

Valeur du type	Signification
<b>0</b>	Réponse à une demande d'écho
<b>3</b>	Destination inaccessible
<b>4</b>	Limitation de production à la source
<b>5</b>	Redirection
<b>8</b>	Demande d'écho
<b>11</b>	Expiration de délai pour un datagramme
<b>12</b>	Problème de paramètre d'un datagramme
<b>13</b>	Demande d'horodatage
<b>14</b>	Réponse à une demande d'horodatage
<b>17</b>	Demande de masque d'adresse
<b>18</b>	Réponse à une demande de masque

Code du message :

Valeur du code	Signification
<b>0</b>	Réseau inaccessible (si type != 0 ou != 8)
<b>1</b>	machine inaccessible
<b>2</b>	Protocole inaccessible
<b>3</b>	Port inaccessible
<b>4</b>	fragmentation nécessaire mais impossible à cause du drapeau DF
<b>5</b>	le routage a échoué
<b>6</b>	réseau inconnu
<b>7</b>	machine inconnue
<b>8</b>	machine non connectée au réseau (inutilisé)
<b>9</b>	communication avec le réseau interdite
<b>10</b>	communication avec la machine interdite
<b>11</b>	réseau inaccessible pour ce service
<b>12</b>	machine inaccessible pour ce service

## VI - Protocole ARP (Address Resolution Protocol)

**Address Resolution Protocol (ARP)**, protocole de résolution d'adresse) est un protocole effectuant la traduction d'une **adresse** de protocole **de couche réseau** (typiquement une adresse **IPv4**) en une **adresse ethernet** (typiquement une adresse **MAC**).

Il est nécessaire au fonctionnement d'**IPv4**, mais semble inutile au fonctionnement d'**IPv6**. En IPv6, ARP devient obsolète et est remplacé par Internet Control Message Protocol V6.

Voici à quoi ressemble un **message ARP** encapsulé dans la trame **Ethernet** :

Bits 0 - 7	Bits 8 - 15	Bits 16 - 31
type de matériel		type de Protocole
longueur de l'adresse physique	longueur de l'adresse logique	Opération
adresse physique (MAC) source (6 octets)		
adresse logique (IP) source (4 octets)		
adresse physique (MAC) de destination (6 octets)		
adresse logique (IP) de destination (4 octets)		

**Type de matériel** (16 bits) :

- **01** - Ethernet (10Mb) [JBP]
- **15** - Frame Relay [AGM]
- **16** - Asynchronous Transmission Mode (ATM) [JXB2]
- **17** - HDLC [JBP]
- **18** - Fibre Channel [Yakov Rekhter]
- **19** - Asynchronous Transmission Mode (ATM) [RFC2225]
- **20** - Serial Line [JBP]
- **21** - Asynchronous Transmission Mode (ATM) [MXB1]

**Type de protocole** (16 bits) : Type de protocole couche 3 (OSI) qui utilise Arp.

- **0x0800** - IP

**Longueur de l'adresse physique** (8 bits) : La longueur doit être prise en octets.

- **01** - Token Ring
- **06** - Ethernet

**Longueur de l'adresse logique** (8 bits) : La longueur doit être prise en octets.

- **04** - IP v4
- **16** - IP v6

**Opération** (16 bits) :

- **01** - Request (*requête*)
- **02** - Reply (*réponse*)