

# Les Réseaux

## les protocoles TCP/IP

*Version 2*

*Auteur : Christophe VARDON  
professeur STI – Bac Pro MRIM  
formateur TICE iufm*

## Table des matières

<b>1. Historique - Applications.....</b>	<b>3</b>
1.1. Les atouts TCP/IP.....	3
1.2. Quelles applications ?.....	3
1.3. Protocole, pilote, interface.....	3
1.4. Version de IP.....	4
1.5. Place dans le modèle OSI.....	4
<b>2. A quoi servent les protocoles TCP/IP ? : fonctionnalités des protocoles IP, TCP, ARP, ICMP.....</b>	<b>5</b>
<b>3. Analyse du protocole IP (Internet Protocol).....</b>	<b>6</b>
3.1. Structure d'une adresse IP.....	6
3.3. Des plages d' adresses réservées!.....	7
3.4. Les domaines et les noms de machine.....	7
3.5. La fonction de routage.....	7
3.6. Les masques de (sous-)réseau.....	8
3.7 Description du datagramme IP :.....	10
<b>4. Analyse des protocoles TCP et UDP (Transfert Control Protocol).....</b>	<b>12</b>
4.1. Fonctionnalités.....	12
4.2. Description du segment .....	12
4.3. Etablissement d'une connection TCP.....	13
4.4. Différences entre TCP et UDP.....	13
4.5. Etude de cas des services WINDOWS XP Pro et 2000 Advanced Server.....	14
<b>5. Analyse du protocole ARP (Address Resolution Protocol) .....</b>	<b>15</b>
<b>6. Analyse du protocole ICMP.....</b>	<b>16</b>

# 1. Historique - Applications

Ce protocole de communication a été mis au point à partir d'une étude commandée au début des années 1970 par le DARPA (Defense Advanced Project Research Agency) dépendant du DoD (Department of Defense) Américain. L'objectif était de mettre au point un protocole de communication permettant d'interconnecter les ordinateurs de toutes marques dont disposait l'armée des US.

Les premières implémentations ont été réalisées au début des années 1980 . Elles introduisaient les notions de : couches de communication. Le protocole TCP/IP ne respecte pas totalement la norme OSI.

## 1.1. Les atouts TCP/IP

Il a été « adopté » très tôt par les systèmes Unix, ce qui lui apporté fiabilité et crédit. Les spécifications sont du domaine public, et elles sont facilement accessibles à tous, ce qui a permis de nombreux développements dans les milieux universitaires et de la recherche. Les spécifications sont fournies sous la forme de RFC (Request for Comments).

Diversité technologique : il est disponible sur la plupart des plates-formes matérielles et systèmes d'exploitation, de l'ordinateur personnel (PC ou Mac) au plus gros calculateur vectoriel (Cray, ... ).

Adaptabilité technique : Il est utilisable sur la plupart des réseaux physiques ( Ethernet 802.3 , Token Ring 802.5, liaisons séries ) et même à travers d'autres réseaux publics ( X25, Numéris).

Diversité logicielle : De très nombreux logiciels ont été développés sur TCP/IP, qu'ils soient du domaine public ou vendus par des sociétés spécialisées.

C'est le principe de : **IP au-dessus de tout** , il fonctionne sur :

Ethernet (RFC894) Token Ring Liaison série de 9,6Kbits/s à 2Mbits/s : SLIP (RFC1055) PPP (Point to Point Protocol)	FDDI (RFC1188) FastEthernet 100Mbps ATM X25
---	--

## 1.2. Quelles applications ?

- telnet : connexion interactive, prise de contrôle à distance.
- ftp ( File Transfert Protocol) : transfert de fichier
- smtp (Simple Mail Tranfert Protocol) : messagerie
- nfs : (Network File System): système de fichiers répartis

**Remarque** : sur un même réseau physique (Ethernet par exemple) le protocole TCP/IP peut cohabiter avec d'autres protocoles de niveau 3 . Pour cela dans la trame de niveau 2 un champ identifie le type de protocole de niveau 3.

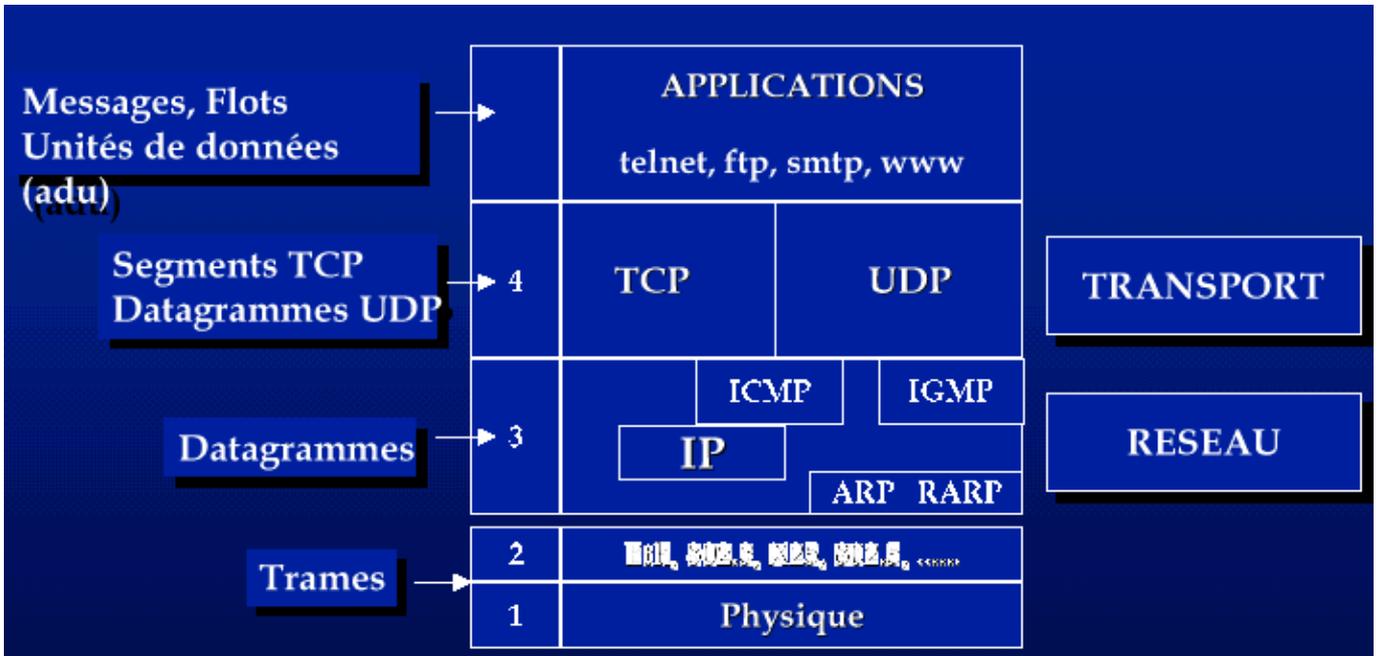
## 1.3. Protocole, pilote, interface

Plusieurs protocoles peuvent même cohabiter sur une même machine : le niveau 2 est géré par le pilote (driver) de la carte (Ethernet par ex), au dessus duquel il y a plusieurs "piles" de niveau supérieur. Le paquet extrait de la trame est transmis à la pile correspondant au type de protocole (cf notion de SAP - Service Access Point - du modèle OSI)

### 1.4. Version de IP

On utilise actuellement la version 4 de TCP/IP (dite : ipv4), et on passera progressivement à la version 6 (dite : ipv6) dans les années à venir .

### 1.5. Place dans le modèle OSI



## 2. A quoi servent les protocoles TCP/IP ? : fonctionnalités des protocoles IP, TCP, ARP, ICMP

**PROBLÈME** : comment identifier la machine destinataire d'un message circulant sur le réseau ?

**(MAUVAISE) RÉPONSE** : l'« adresse » MAC de la carte réseau n'est pas suffisante car elle ne donne aucune indication sur le réseau dont fait partie la machine : on est limité au réseau local;

**(BONNE) RÉPONSE** : Il faut une adresse qui identifie la machine elle-même, mais aussi le réseau (LAN) dont elle fait partie.

- le protocole IP autorise l'attribution d'une adresse de type *réseau.machine* à chaque poste informatique.
- l'adresse IP permet d'identifier le réseau et la machine destinataire d'un message.
- Conclusion : le protocole IP a pour rôle d'identifier la source et le destinataire d'une trame et de fournir une route pour son acheminement. (fonction de routage)

**Oui, mais** : sur la machine destinataire peuvent tourner plusieurs logiciels; par exemple « *Internet Explorer* » pour naviguer sur le Web et « *Outlook Express* » pour le courrier électronique.

- **PROBLÈME** : comment le système d'exploitation va-t-il savoir à quelle application sont destinées les données reçues ?
- **RÉPONSE** : le protocole TCP rajoute à la trame un numéro (appelé « port TCP ») qui indique l'application destinataire des données (ex : le port 80 pour le web/http)
- Un **socket** est une adresse réseau constituée par la concaténation d'une adresse Internet avec un numéro de port TCP (ex : **192.168.1.2:80**).

### « Well known ports »

Les numéros de port standards sont définis la **RFC1700**; complétez le tableau suivant :

<i>Application</i>	<i>n° du port TCP</i>	<i>Application</i>	<i>n° du port TCP</i>
Web - http (ex : IE5.5)		Web sécurisé - https	
Mail sortant - smtp (ex : Outlook)		courrier entrant - pop	
proxy http (ex : squid proxy serv)		partage de fichier - netbios	
administration distante - snmp		Transfert de fichier - ftp	
telnet (remote terminal)		SSH – telnet sécurisé	

### 3. Analyse du protocole IP (Internet Protocol)

L'adresse IP d'une machine est une adresse de niveau réseau codée sur 32 bits (ie 4 octets en IPv4) qui est en général notée sous la forme de 4 chiffres séparés par des points. On parle de notation en décimal pointé. Chaque champ, qui représente un octet, peut prendre des valeurs entre 0 et 255.

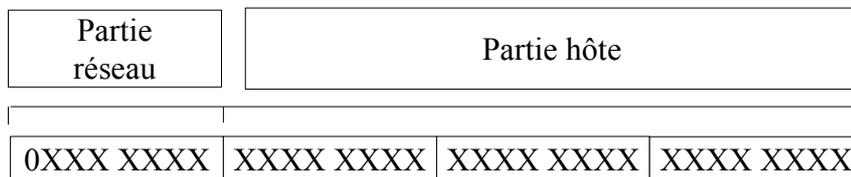
Exemple : 192.93.116.3

#### 3.1. Structure d'une adresse IP

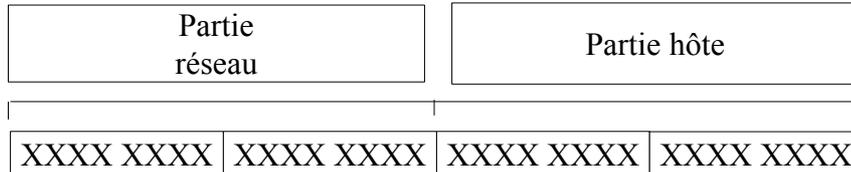
L'adresse IP est constituée d'un champ *numéro de réseau* (1, 2 ou 3 octets) et d'un champ *numéro de machine dans le réseau* (3, 2 ou 1 octets). L'adresse ip = adresse de réseau + adresse de machine.

Les réseaux TCP/IP sont rangés en 3 classes A, B ou C :

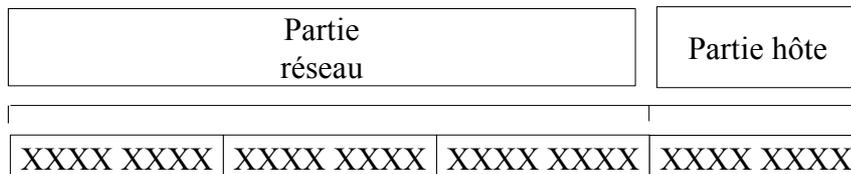
- classe A : 1 à 127 .X.X.X



- classe B : 128 à 191 .X.X.X



- classe C : 192 à 223 .X.X.X ( les adresses > à 223 sont réservées à d'autres usages)



Le nombre de machines dans le réseau dépend donc de la classe du réseau. Chaque octet du champ machine peut prendre des valeurs entre 1 et 254. Les valeurs 0 (tous les bits à 0) et 255 (tous les bits à 1) sont réservées :

- Un champ machine tout à 0 sert à désigner le numéro de réseau (notamment pour le routage)
- Un champ tout à 1 indique un message de broadcast adressé à toutes les machines IP du réseau.

### 3.3. Des plages d' adresses réservées!

0.0.0.0	est réservée pour la route par défaut. L'adresse désigne tous les réseaux. Tous les paquets destinés à un réseau inconnu, seront dirigés vers cette route.
127.0.0.0	est réservée au trafic IP de la machine locale. Une interface locale porte en générale l'adresse 127.0.0.1 appelée adresse de "loopback" ou boucle locale.
10.X.X.X	plages d'adresses peuvent être librement utilisées pour monter un réseau privé
172.16.X.X 172.31.X.X	plages d'adresses peuvent être librement utilisées pour monter un réseau privé
192.168.X.X	plages d'adresses peuvent être librement utilisées pour monter un réseau privé

*Note : aucun paquet provenant d'un réseau privé ou à destination d'un réseau privé, ne peut (ou ne doit...) être routé sur l'internet.*

### 3.4. Les domaines et les noms de machine

- Il est peu commode de désigner une machine par son adresse IP.
- L'utilisateur « humain » utilise un nom qui se présente sous la forme :  
*nom\_machine.sous\_domaine.domaine* (ex : www.google.fr).
- Malgré tout, c'est l'adresse IP chiffrée qui est utilisée en interne dans les paquets au cours des échanges. Il faut donc un mécanisme qui permette de traduire le nom\_machine en adresse IP.
- Des ordinateurs appelés « Serveurs de noms » ou « DNS » se chargent de cette traduction

### 3.5. La fonction de routage

Les réseaux IP sont interconnectés par des « routeurs » IP (parfois appelés « passerelles »). Chaque station IP doit connaître le routeur par lequel il faut sortir pour pouvoir atteindre un réseau extérieur.

Un routeur dispose de plusieurs interfaces réseau et contient une table qui lui indique sur quelle interface tel ou tel réseau est relié. Il oriente donc **physiquement** la trame sur la bonne **route**!

#### Exemple :

- Réseau 1 --> Interface Ethernet 1
- Réseau 2 --> Interface Ethernet 2
- Autres réseau --> Interface Modem

Les tables de routage peuvent être statiques dans le cas de réseaux simples, ou dynamiques dans le cas de réseaux maillés. Le protocole d'échange dynamique des tables IP sur un réseau local est *RIP* ( Routing Information Protocol) ou le protocole OSPF.

### 3.6. Les masques de (sous-)réseau

#### Les masques de réseau

Pour que le réseau Internet puisse router (acheminer) les paquets de données, il faut qu'il connaisse l'adresse IP du réseau local de destination. Pour la calculer à partir de l'adresse IP de destination, on utilise le masque de sous réseau.

A chaque classe d'adresses est associé un masque de réseau, ou netmask, qui est constitué de 32 bits. Le tableau suivant fournit les différents masques pour les trois classes traditionnelles.

Classe	Masque
A	255. 0. 0. 0
B	255. 255. 0. 0
C	255. 255. 255. 0

Un « ET » logique appliqué entre le masque de réseau et l'adresse IP permet d'obtenir l'adresse d'un réseau correspondant.

Calcul de l'adresse réseau en décimal

@ IP	193	252	19	3
Masque Réseau	255	255	255	0
@ Réseau	193	252	19	0

Calcul de l'adresse réseau en binaire

@ IP	1100 0001	1111 1100	0001 0011	0000 0011
Masque Réseau	1111 1111	1111 1111	1111 1111	0000 0000
@ Réseau	1100 0001	1111 1100	0001 0011	0000 0000

Ainsi, à l'aide du masque de réseau, on peut définir, pour toute adresse IP :

L'adresse réseau associée, la partie hôte associée, l'adresse de diffusion associée qui désigne tous les hôtes de ce réseau.

Le tableau suivant fournit ces informations pour trois adresses IP prises parmi les trois classes fondamentales.

Adresse IP	10. 25. 2. 5	172. 17. 5. 8	192. 168. 53. 24
Classe	A	B	C
Masque de réseau	255. 0. 0. 0	255. 255. 0. 0	255. 255. 255. 0
Adresse de réseau	10. 0. 0. 0	172. 17. 0. 0	192. 168. 53. 0
Adresse de diffusion	10. 255. 255. 255	172. 17. 255. 255	192. 168. 53. 255
Complément à 1 du masque	0.255.255.255	0.0.255.255	0.0.0.255
Partie hôte de l'adresse	0.25.2.5	0.0.5.8	0.0.0.24

*Note* : les classes d'adresses sont utilisées dans les réseaux locaux, mais depuis les années 90, elles ne sont plus utilisées par les routeurs internet (CIDR : Classless Internet Domain Routing)

## Les masques de sous - réseaux

Parfois, on est amené à répartir les adresses IP d'un même réseau de classe A, B ou C sur plusieurs supports physiques. Par exemple, si on dispose d'une cinquantaine de machines, à répartir sur trois réseaux Ethernet par exemple, il serait inutilement coûteux d'acheter trois réseaux de classe C : une seule classe C peut déjà accueillir 254 machines.

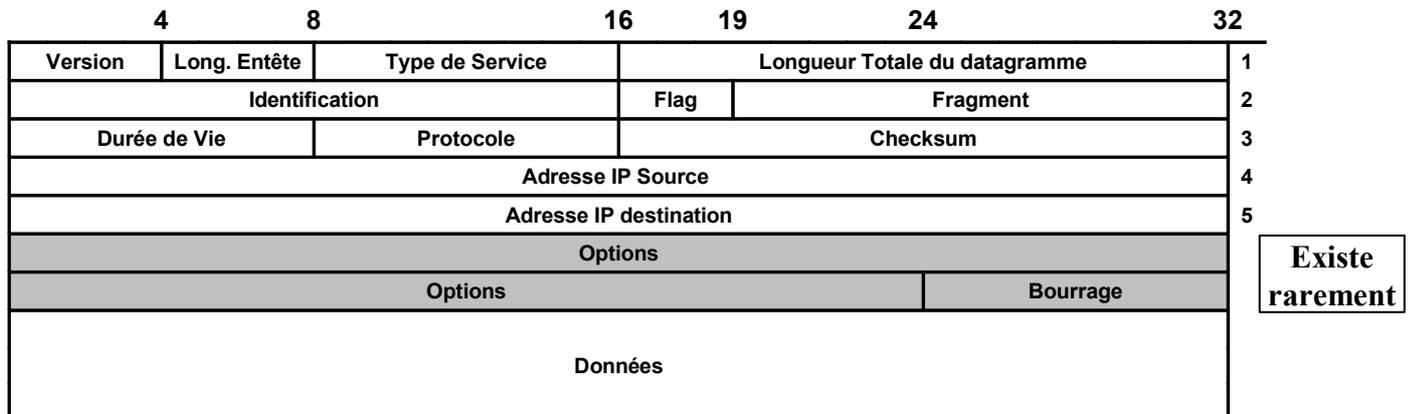
→ Pour résoudre ce problème, il faut introduire un nouveau type de masque : le masque de sous - réseaux.

Le principe est simple : le réseau est découpé en sous - réseaux de même taille. Pour cela, **la partie hôte des adresses est elle-même découpée en deux plages de bits** :

la plage des bits de poids forts correspond aux bits identifiant les sous réseaux  
l'autre plage désigne le numéro de machine dans le sous réseau.

### 3.7 Description du datagramme IP :

La structure générale d'un datagramme IP est représentée sur la figure suivante :



#### Version (Vers)

Sur ces 4 bits est codé le numéro de version du protocole IP utilisé. Actuellement, il s'agit pratiquement toujours de la version **4 (0100)**. Le numéro **5** sert à des applications expérimentales et la version **6** est en cours de mise en service (celle-ci utilise une autre structure de datagramme).

#### Longueur d'entête (I.H.L.)

Comme la longueur d'une entête de datagramme IP est variable, elle est codée sur 4 bits, sous forme de mots de 32 bits (4 octets). (Par exemple, **5** représente 5 fois 32 bits, soit **20 octets**).

#### Type de service (Type of Service ToS)

Ce champ de *8 bits* possède la structure suivante : *(Pour l'instant une seule option peut être sélectionnée)*

Priorité			Type de Service				
			-délai	+débit	+fiable	-coût	0
0	1	2	3	4	5	6	7

- 3 bits indiquant la priorité.
  - 000 (Par défaut - priorité la moins élevée)
  - 111 (supervision réseau- priorité la plus élevée). Cette partie est utilisée par certaines passerelles.
- 4 bits indiquant le type de service souhaité (en fonctionnement normal, ils sont tous à zéro) :
  - Le premier bit demande au routeur de choisir un chemin ayant un délai de transmission le plus court possible (par exemple choisir de passer par un câble sous marin plutôt que par une liaison satellite)
  - Le deuxième bit demande au routeur un débit élevé.
  - Le troisième bit demande au routeur de diriger les paquets vers des liaisons fiables
  - Le quatrième bit demande au routeur de choisir un chemin ayant un coût minimum.
- 1 bit réservé pour le futur, devant actuellement rester à zéro sauf pour certains cas expérimentaux.

#### Longueur du datagramme (Long. Totale)

Longueur totale du datagramme (entête + données), en octets. Comme ce champ est codé sur 16 bits, la longueur maximale d'un datagramme IP est de 65535 octets. **Attention** : si un datagramme est fragmenté, ce champ fait référence à la longueur du fragment courant et non à la longueur du datagramme initial.

**Fragmentation**

Si le routeur reçoit un datagramme trop grand pour le support sur lequel il doit l'envoyer, il est obligé de fragmenter ce datagramme. Il est donc nécessaire de repérer ces fragments et de faire en sorte que le récepteur des fragments d'un datagramme puisse « recoller » les morceaux dans le bon ordre pour reconstituer le datagramme. 3 champs permettent de résoudre ce problème.

**Identification (ID)**

Ce champ sur 16 bits sert à identifier un datagramme ou les fragments d'un datagramme fragmenté. En effet, ce champ a *la même valeur pour tous les fragments provenant d'un même datagramme*.

**Drapeau (Flg)**

Ces 3 bits donnent des informations concernant la fragmentation :

0	DF	MF
---	----	----

- ✓ Le premier n'est pas utilisé et doit rester à zéro.
- ✓ Le deuxième sert à *interdire la fragmentation* du datagramme (0 - Autorisé, 1 - Interdit) (appelé bit de « non-fragmentation » ou « Don't Fragment » DF).
- ✓ Le troisième est à zéro s'il s'agit du *dernier fragment* du datagramme et à un si d'autres fragments doivent encore arriver (appelé bit « fragments à suivre » ou « More Fragment » MF).

**Place du fragment (Fragment)**

Ce nombre codé sur 13 bits indique la position qu'a le 1<sup>er</sup> octet de donnée du fragment dans le datagramme (non fragmenté). **Ce nombre est un multiple de 8 octets**. Ainsi, une valeur de 10 signifierait que le 1<sup>er</sup> octet de donnée de ce fragment est en fait l'octet 80 du datagramme non fragmenté. S'il s'agit du premier fragment, ou si le datagramme n'est pas fragmenté, tous les bits sont à zéro.

**Durée de vie (Time to Live TTL)**

Théoriquement, ce champ doit indiquer sur 8 bits le nombre de secondes pendant lequel le datagramme est autorisé à voyager. A chaque passage d'une passerelle, on retire le temps qu'a pris la traversée de la passerelle. En pratique, comme ce temps est souvent inférieur à une seconde, on retire 1, ce qui fait que ce champ indique plus souvent le nombre de passerelles par lesquelles le datagramme peut passer qu'une durée proprement dite. Quand le champ durée de vie atteint zéro, le datagramme est détruit et un message d'erreur est envoyé à l'émetteur. Cette méthode permet d'éviter qu'un datagramme ne circule indéfiniment en boucle.

**Numéro de protocole (Protocole)**

Ce champ de 8 bits indique à quel protocole de niveau plus élevé est destiné le datagramme (SAP). La valeur zéro est réservée. Quelques exemples de valeurs : (La liste complète se trouve dans la RFC 790)

- 1 : ICMP (Internet Control Message Protocol)
- 6 : TCP (Transmission Control Protocol)
- 17 : UDP (User Datagram Protocol)

**Adresse IP source**

Adresse IP de l'émetteur du datagramme.

**Adresse IP destination**

Adresse IP du destinataire du datagramme.

**Données**

Ce sont les données du datagramme proprement dites.

**Résumé :**

Le protocole Internet est responsable de **l'adressage** et du **roulage** entre machines, du cheminement des paquets de données dans le réseau, de la constitution et du réassemblage des paquets. Les fonctionnalités assurées par IP peuvent se déduire de l'examen de l'en-tête du paquet. Il identifie entre autres la source et la destination du paquet et comporte des identificateurs de fragmentation.



**Réservé: 6 bits**

Réservés pour usage futur. Doivent nécessairement être à 0.

**Bits de contrôle: 6 bits (de gauche à droite):**

- URG: Pointeur de données urgentes significatif
- ACK: Accusé de réception significatif
- PSH: Fonction Push
- RST: Réinitialisation de la connexion
- SYN: Synchronisation des numéros de séquence
- FIN: Fin de transmission

**Fenêtre: 16 bits**

Le nombre d'octets à partir de la position marquée dans l'accusé de réception que le récepteur est capable de recevoir.

**Checksum: 16 bits**

La somme de contrôle

**Pointeur de données urgentes: 16 bits**

Communique la position d'une donnée urgente en donnant son décalage par rapport au numéro de séquence. Le pointeur doit pointer sur l'octet suivant la donnée urgente. Ce champs n'est interprété que lorsque URG est marqué.

**Options: variable**

Les champs d'option peuvent occuper un espace de taille variable à la fin de l'en-tête TCP. Ils formeront toujours un multiple de 8 bits. Un octet d'option commence toujours sur un nouvel octet.

**Bourrage (padding): variable**

Les octets de bourrage terminent l'en-tête TCP:

- de sorte que le nombre d'octet de celle-ci soit toujours multiple de 4 (32 bits)
- de sorte que l'offset de données marqué dans l'en-tête corresponde bien au début des données applicatives.

### 4.3. Etablissement d'une connexion TCP

Considérons la capture de trame effectuée avec *Ethereal* :

No.	Time	Source	Destination	Protocol	Info
10	10.181832	192.168.0.10	212.27.35.1	TCP	4252 > http [SYN]
11	10.204707	212.27.35.1	192.168.0.10	TCP	http > 4252 [SYN, ACK]
12	10.204848	192.168.0.10	212.27.35.1	TCP	4252 > http [ACK]
13	10.205333	192.168.0.10	212.27.35.1	HTTP	GET / HTTP/1.1

L'établissement d'une connexion TCP suit un protocole strict :

- Une requête de synchronisation [SYN] de la part de l'initiateur du dialogue (le client),
- une réponse d'accusé réception de la synchronisation [SYN,ACK] de la part du serveur,
- un accusé réception du client [ACK]

### 4.4. Différences entre TCP et UDP

TCP est un protocole beaucoup plus complexe qu'UDP, il se charge, entre autres, de remettre en ordre, avant leur délivrance, les paquets qui lui parviennent.

Le protocole UDP (User Datagram Protocol) est un protocole de transmission de datagrammes sur le réseau qui fournit de manière optionnelle un certain nombre de contrôles. Un datagramme est un paquet de données considéré comme une entité isolée et indépendante, c'est-à-dire qu'il comporte dans son en-tête toutes les informations nécessaires à son acheminement à travers le réseau jusqu'à son destinataire.

La transmission de paquets composant le message est donc assurée de manière totalement indépendante pour chaque paquet. On pourrait, dans une certaine mesure, comparer ce type de service au service postal qui prend en charge les messages et assure leur transport à destination mais sans garantir le chemin parcouru par chaque message, ni le temps mis pour le parcourir, ni à fortiori le respect d'une séquentialité dans la délivrance des messages.

## 4.5. Etude de cas des services *WINDOWS XP Pro* et *2000 Advanced Server*

On utilise un outil de « hacker » pour lister les ports ouverts sur une machine *Windows* : cela nous permet de déterminer les services qui « tournent »; L'outil « **Nmap** » donne les résultats suivants sur un *Windows 2000 Advanced Server* :

```
Starting nmap v. 2.54BETA25 ( www.insecure.org/nmap/ )
Interesting ports on fenetre-w2k.hsc.fr (192.70.106.143):
(The 65524 ports scanned but not shown below are in state:
closed)
Port State Service
25/tcp open smtp
80/tcp open http
135/tcp open loc-srv
139/tcp open netbios-ssn
443/tcp open https
445/tcp open microsoft-ds
1025/tcp open listen
1026/tcp open nterm
1027/tcp open unknown
3372/tcp open unknown
4983/tcp open unknown
Nmap run completed -- 1 IP address (1 host up) scanned in
115 seconds
nmap -sU 192.70.106.143 -p 1-65535
Starting nmap v. 2.54BETA25 ( www.insecure.org/nmap/ )
Interesting ports on fenetre-w2k.hsc.fr (192.70.106.143):
(The 65527 ports scanned but not shown below are in state:
closed)
Port State Service
135/udp open loc-srv
137/udp open netbios-ns
138/udp open netbios-dgm
445/udp open microsoft-ds
500/udp open isakmp
1028/udp open unknown
```

On fait de même avec une machine *XP Pro*; Finalement après avoir vérifié les applications correspondant à ces ports, voici les services ouverts en standard sous *2000* et *XP* :

Service	Port (s)	Service(s) Windows
Serveur IIS	25/tcp 80, 443/tcp 3456/udp >1024/tcp, >1024/udp <4000-5000>/tcp	<i>SMTP (Simple Mail Transfert Protocol)</i> <i>World Wide Web Publishing Service</i> <i>IIS Admin Service ?</i> 2 ports dynamiques (RPC) 1 port (site d'administration)
Protocole IKE	500/udp	<i>IPSEC Policy Agent</i>
NetBIOS sur TCP	137/udp, 138/udp, 139/tcp	<i>TCP/IP NetBIOS Helper Service</i>
CIFS/SMB	139/tcp, 445/tcp	<i>Server, Workstation</i>
<i>Portmapper</i> RPC	135/tcp, 135/udp	<i>Remote Procédure Call (RPC)</i>
MSDTC	3372/tcp, >1024/tcp	<i>Distributed Transaction Coordinator</i>
Tâches programmées	>1024/tcp	<i>Task Scheduler</i>
Messenger	>1024/udp	<i>Messenger</i>

FIG. 1 - Services réseaux sur un système Windows 2000 serveur

Service	Port(s)	Service(s) Windows
Protocole IKE	500/udp	<i>IPSEC Policy Agent</i>
Protocole NTP	123/udp	<i>Windows Time</i>
UPnP	5000/tcp, 1900/udp	<i>SSDP Discover-y Service</i>
Cache DNS	>1024/udp	<i>DNS Client</i>
NetBIOS sur TCP	137/udp, 138/udp, 139/tcp	<i>TCP/IP NetBIOS Helper Service</i>
CIFS/SMB	139/tcp, 445/tcp	<i>Server, Workstation</i>
<i>Portmapper</i> RPC	135/tcp, 135/udp	<i>Remote Procédure Call (RPC)</i>
Tâches programmées	>1024/tcp	<i>Task Scheduler</i>
Messenger	>1024/udp	<i>Messenger</i>

FIG. 2 - Services réseaux sur un système Windows XP professionnel

## 5. Analyse du protocole ARP (Address Resolution Protocol)

C'est un protocole *de résolution (~traduction) d'adresse à l'intérieur du réseau local (LAN)*

### L'adressage à l'intérieur d'un réseau local (~Ethernet!) se fait grâce à l'adresse MAC

> Ethernet ne connaît pas les adresses IP! aïe !

> Il est nécessaire d'établir **un lien entre l'adresse IP d'une station et son adresse MAC** pour qu'un paquet IP arrivant dans un réseau local puisse être acheminé vers la bonne station. C'est le protocole ARP qui va permettre d'établir ce lien.

ARP traduit des adresses de type INTERNET sur 32 bits en adresses ETHERNET sur 48 bits. ARP se présente comme **un service** qui gère des **tables de correspondance d'adresses** et répond à des requêtes d'identification. Lorsqu'il reçoit une requête, il crée un **message de demande ARP** : « **qui possède l'adresse IP x.x.x.x ?** »; qui est diffusé (broadcast) à tout le réseau en attente d'une réponse positive de la part d'une des machines connectées. Une réponse positive provoque une mise à jour par ARP de ses tables de traduction d'adresses.

Par exemple : un équipement A veut envoyer un datagramme IP à un équipement B qui est connecté sur le même LAN; « Fatalement », il doit l'encapsuler dans une trame de la couche MAC (en général Ethernet-802.3). Le problème est que A ne connaît à priori que l'adresse IP de B et pas son adresse MAC. Il faut donc un protocole qui permette de découvrir l'adresse MAC de B à partir de son adresse IP. C'est le rôle du protocole ARP.

Voici quelques trames capturées avec Ethereal :

No.	Time	Source	Destination	Protocol	Info
1	0.000000	00:50:ba:eb:32:10	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.2.2? Tell 192.168.2.1
2	121353	00:40:f4:1c:d8:e4	00:50:ba:eb:32:10	ARP	192.168.2.2 is at 00:40:f4:1c:d8:e4
46	20.961	Microsoft_54:00:00	Broadcast	ARP	Who has 213.36.80.1? Tell 213.36.24.90

→ Commentez cette trame

→ Capturez une demande et une réponse ARP avec Ethereal puis imprimez le détail de celle-ci. Commentez chaque ligne.

## 6. Analyse du protocole ICMP

Le protocole ICMP (Internet Control Message Protocol) constitue le protocole des messages d'erreur. Les messages ICMP sont classés en plusieurs catégories :

La première est constituée de tous les messages résultant d'un incident réseau, où qu'il se soit produit, et qui peuvent être transmis à l'émetteur du paquet ayant subi l'incident. On peut classer dans cette catégorie les erreurs de routage ou celles qui résultent d'une indisponibilité du destinataire.

La seconde classe est constituée de tous les messages d'erreur induits par des incidents entre une machine hôte et la porte (gateway) par laquelle passent les paquets, par exemple, il peut s'agir d'une procédure de routage qui informe l'hôte d'un meilleur chemin que celui qui a été choisi à l'origine.

La dernière catégorie concerne tout ce qui est gestion de réseau, les tests de connexion, les mesures de performances et de trafic (ping).

Toutes les actions, transmissions ou redirections induites par un message ICMP sont prises en charge par la couche ICMP de IP.

Voici quelques trames capturées avec Ethereal lors d'une commande « *ping 192.168.2.2* »:

No.	Time	Source	Destination	Protocol	Info
5	0.233602	192.168.2.1	192.168.2.2	ICMP	Echo (ping) request
6	121353.49	192.168.2.2	192.168.2.1	ICMP	Echo (ping) reply

→ Commentez cette trames

→ Capturez une demande et une réponse ARP avec Ethereal puis imprimez le détail de celle-ci. Commentez chaque ligne.