Cours TCP/IP

Cours TCP/IP

Ensemble de documents réalisés pour la freeduc-sup.

Historique des versions

Version 1.14 2004/10/31 01:20:26

Table des matières

Liste des illustrations

Chapitre 1. Eléments de cours sur TCP/IP

La suite de protocoles TCP/IP

Le document présente la suite de protocoles TCP/IP.

Ce document sert d'introduction à l'ensemble des cours et TP sur les différents protocoles

1.1. Présentation de TCP/IP

TCP/IP est l'abréviation de Transmission Control Protocol/Internet Protocol. Ce protocole a été développé, en environnement UNIX, à la fin des années 1970 à l'issue d'un projet de recherche sur les interconnexions de réseaux mené par la DARPA (Defense Advanced Research Projects Agency) dépendant du DoD (Department of Defense) Américain.

TCP/IP ,devenu standard de fait, **est actuellement la famille de protocoles réseaux qui gère le routage la plus répandue** sur les systèmes Unix et Windows, et surtout, c'est le protocole de l'Internet.

Plusieurs facteurs contribuent à sa popularité :

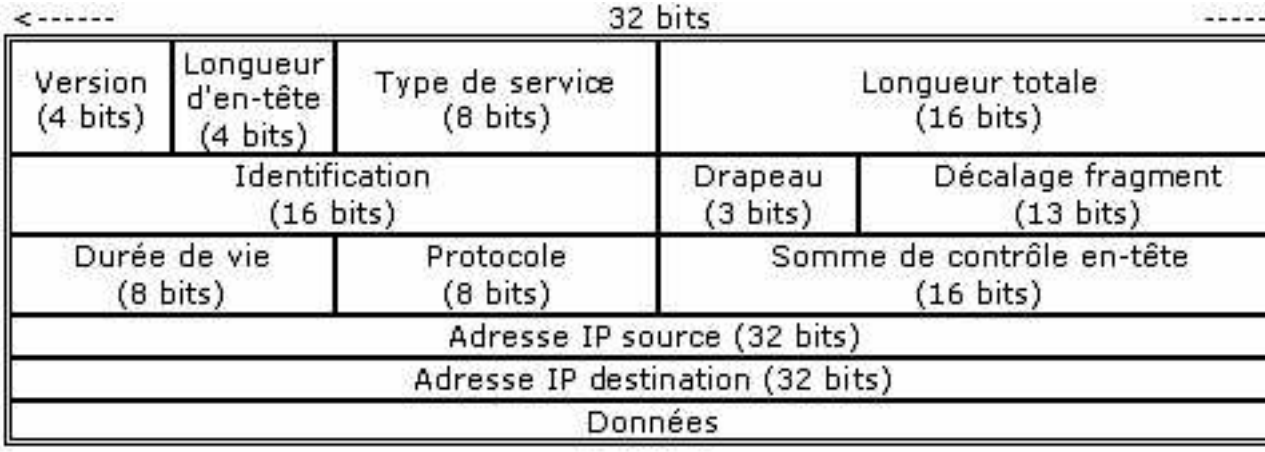
Maturité, Ouverture, Absence de propriétaire, Richesse (il fournit un vaste ensemble de fonctionnalités), Compatibilité (différents systèmes d'exploitation et différentes architectures matérielles), et le développement important d'Internet.

La famille des protocoles TCP/IP est appelée **protocoles Internet**, et a donné son nom au réseau du même nom. Leurs spécifications sont définies dans des documents du domaine public appelés RFC (Request For Comments - Appels à commentaires). Ils sont produits par l'IETF (Internet Engineering Task Force) au sein de l'IAB (Internet Architecture Board).

La RFC 826, par exemple, définit le protocole ARP.

Le datagramme correspond au format de paquet défini par le protocole Internet. Les cinq ou six (sixième facultatif) premier mots de 32 bits représentent les informations de contrôle appelées en-tête.

Figure 1-1. datagramme IP



La longueur théorique maximale d'un datagramme IP est de 65535 octets. En pratique la taille maximale du datagramme est limitée par la longueur maximale des trames transportées sur le réseau physique. La fragmentation du datagramme (définie dans le 2ème mot de 32 bits) devient alors nécessaire dès que sa taille ne lui permet plus d'être directement transporté dans une seule trame physique. Les modules internet des équipements prennent en charge le découpage et le réassemblage des datagrammes.

Le protocole Internet transmet le datagramme en utilisant l'adresse de destination contenue dans le cinquième mot de l'en-tête. L'adresse de destination est une adresse IP standard de 32 bits permettant d'identifier le réseau de destination et la machine hôte connectée à ce réseau.

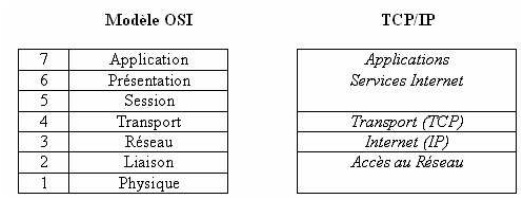
Dans un réseau TCP/IP, on assigne généralement un nom à chaque hôte. Le terme d'hôte est pris dans son sens large, c'est à dire un "noeud de réseau". Une imprimante, un routeur, un serveur, un poste de travail sont des noeuds qui peuvent avoir un nom d'hôte, s'ils ont une adresse IP.

1.2. OSI et TCP/IP

Bien que le protocole TCP/IP ait été développé bien avant que le modèle OSI apparaisse, ils ne sont pas totalement incompatibles. L'architecture OSI est définie plus rigoureusement, mais ils disposent tous deux d'une architecture en couches.

Les protocoles **TCP** et **IP** ne sont que deux des membres de la suite de protocoles TCP/IP qui constituent le **modèle DOD** (modèle en 4 couches). Chaque couche du modèle TCP/IP correspond à une ou plusieurs couches du modèle OSI (*Open Systems Interconnection*) défini par l'ISO (*International Standards Organization*) :

Figure 1-2. OSI et TCP/IP



Des relations étroites peuvent être établies entre la couche réseau et IP, et la couche transport et TCP.

TCP/IP peut utiliser une grande variété de protocoles en couche de niveau inférieur, notamment X.25, Ethernet et Token Ring. En fait, TCP/IP a été explicitement conçu sans spécification de couche physique ou de liaison de données car le but était de faire un protocole adaptable à la plupart des supports.

1.3. La suite de protocoles TCP / IP

Les protocoles TCP/IP se situent dans un modèle souvent nommé "famille de protocoles TCP/IP".

Les protocoles **TCP** et **IP** ne sont que deux des membres de la suite de protocoles IP.

1.3.1. IP (*Internet Protocol*, Protocole Internet)

IP est un protocole qui se ***charge de l'acheminement des paquets*** *pour tous les autres protocoles de la famille TCP/IP.* Il fournit un système de remise de données optimisé sans connexion. Le terme « optimisé » souligne le fait qu'il ne garantit pas que les paquets transportés parviennent à leur destination, ni qu'ils soient reçus dans leur ordre d'envoi. . La fonctionnalité de somme de contrôle du protocole ne confirme que l'intégrité de l'en-tête IP. Ainsi, seuls les protocoles de niveau supérieur sont responsables des données contenues dans les paquets IP (et de leur ordre de réception).

Le protocole IP travaille en **mode non connecté**, c'est-à-dire que les paquets émis par le niveau 3 sont **acheminés de manière autonome** (datagrammes), sans garantie de livraison.

1.3.2. TCP (*Transmission Control Protocol,* Protocole de contrôle de la transmission)

TCP est probablement le protocole IP de niveau supérieur le plus répandu. ***TCP fournit un service sécurisé de remise des paquets.*** *TCP fournit un protocole fiable, orienté connexion****, au-dessus d'IP*** *(ou*

encapsulé à l'intérieur d'IP). TCP garantit l'ordre et la remise des paquets, il vérifie l'intégrité de l'en-tête des paquets et des données qu'ils contiennent. TCP est responsable de la retransmission des paquets altérés ou perdus par le réseau lors de leur transmission. Cette fiabilité fait de TCP/IP un protocole bien adapté pour la transmission de données basée sur la session, les applications client-serveur et les services critiques tels que le courrier électronique.

La fiabilité de TCP a son prix. Les en-têtes TCP requièrent l'utilisation de bits supplémentaires pour effectuer correctement la mise en séquence des informations, ainsi qu'un total de contrôle obligatoire pour assurer la fiabilité non seulement de l'en-tête TCP, mais aussi des données contenues dans le paquet. Pour garantir la réussite de la livraison des données, ce protocole exige également que **le destinataire accuse réception des données**.

Ces accusés de réception (ACK) génèrent une activité réseau supplémentaire qui diminue le débit de la transmission des données au profit de la fiabilité. Pour limiter l'impact de cette contrainte sur la performance, la plupart des hôtes n'envoient un accusé de réception que pour un segment sur deux ou lorsque le délai imparti pour un ACK expire.

Sur une connexion TCP entre deux machines du réseau, les messages (ou paquets TCP) sont acquittés et délivrés en séquence.

1.3.3. UDP (*User Datagram Protocol*)

UDP est un complément du protocole TCP qui offre un **service de datagrammes sans connexion** qui ne garantit ni la remise ni l'ordre des paquets délivrés. Les sommes de contrôle des données sont facultatives dans le protocole UDP. Ceci permet d'échanger des données sur des réseaux à fiabilité élevée sans utiliser inutilement des ressources réseau ou du temps de traitement. Les messages (ou paquets UDP) sont transmis de manière autonome (sans garantie de livraison.).

Le protocole UDP prend également en charge l'envoi de données d'un unique expéditeur vers plusieurs destinataires.

Ex: TFTP(trivial FTP) s'appuie sur UDP, NT4 utilise UDP pour les Broadcast en TCP-Ip

1.3.4. ICMP (*Internet Control Message Protocol*)

ICMP : protocole de messages de contrôle, est un **protocole de maintenance**. Il permet à deux systèmes d'un réseau IP de partager des informations d'état et d'erreur. Utilisé pour les tests et les diagnostics

La commande **ping** utilise les paquets ICMP de demande d'écho et de réponse en écho afin de déterminer si un système IP donné d'un réseau fonctionne. C'est pourquoi l'utilitaire **ping** est utilisé pour diagnostiquer les défaillances au niveau d'un réseau IP ou des routeurs.

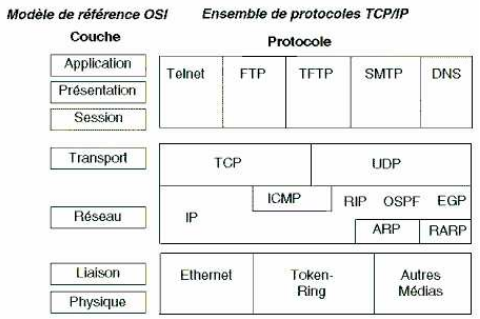
1.3.5. RIP *(Routing Information Protocol)*

RIP est un protocole de routage dynamique qui permet l'échange d'informations de routage sur un inter-réseau. Chaque routeur fonctionnant avec RIP échange les identificateurs des réseaux qu'il peut atteindre, ainsi que la distance qui le sépare de ce réseau *(nb de sauts=nb de routeurs à traverser).* Ainsi chacun dispose de la liste des réseaux et peut proposer le meilleur chemin.

1.3.6. ARP (*Address Resolution Protocol*)

Le protocole ARP permet de déterminer l'adresse physique (ou MAC) d'un noeud à partir de son adresse IP en effectuant une diffusion du type "*qui est X2.X2.X2.X2 ? "*

Figure 1-3. Protocoles TCP/IP et OSI



1.3.7. Fonctionnement général

Pour désigner les informations transmises et leur enveloppe, selon le niveau concerné, on parle de ***message****(ou de flux) entre applications, de* ***datagramme*** *(ou segment) au niveau TCP, de* ***paquet*** *au niveau IP, et enfin, de* ***trames*** *au niveau de l'interface réseau (Ethernet ou Token Ring).*

Les protocoles du niveau application les plus connus sont :

- **HTTP** (Hyper Text Transfer Protocol) permet l'accès aux documents HTML et le transfert de fichiers depuis un site WWW
- **FTP** (File Transfer Protocol) pour le transfert de fichiers s'appuie sur TCP et établit une connexion sur un serveur FTP
- **Telnet** pour la connexion à distance en émulation terminal, à un hôte Unix/Linux.
- **SMTP** (Simple Mail Transfer Protocol) pour la messagerie électronique (UDP et TCP)

- **SNMP** (Simple Network Management Protocol) pour l'administration du réseau
- **NFS** (Network File System) pour le partage des fichiers Unix/Linux.

1.4. Les applications TCP-IP

1.4.1. Modèle client/serveur

Les applications réseaux fonctionnent sur le modèle client/serveur. Sur la machine serveur un processus serveur (daemon) traite les requêtes des clients. Client et serveur dialoguent en échangeant des messages qui contiennent des requêtes et des réponses.

Prenons par exemple telnet.

Figure 1-4. Exemple Telnet

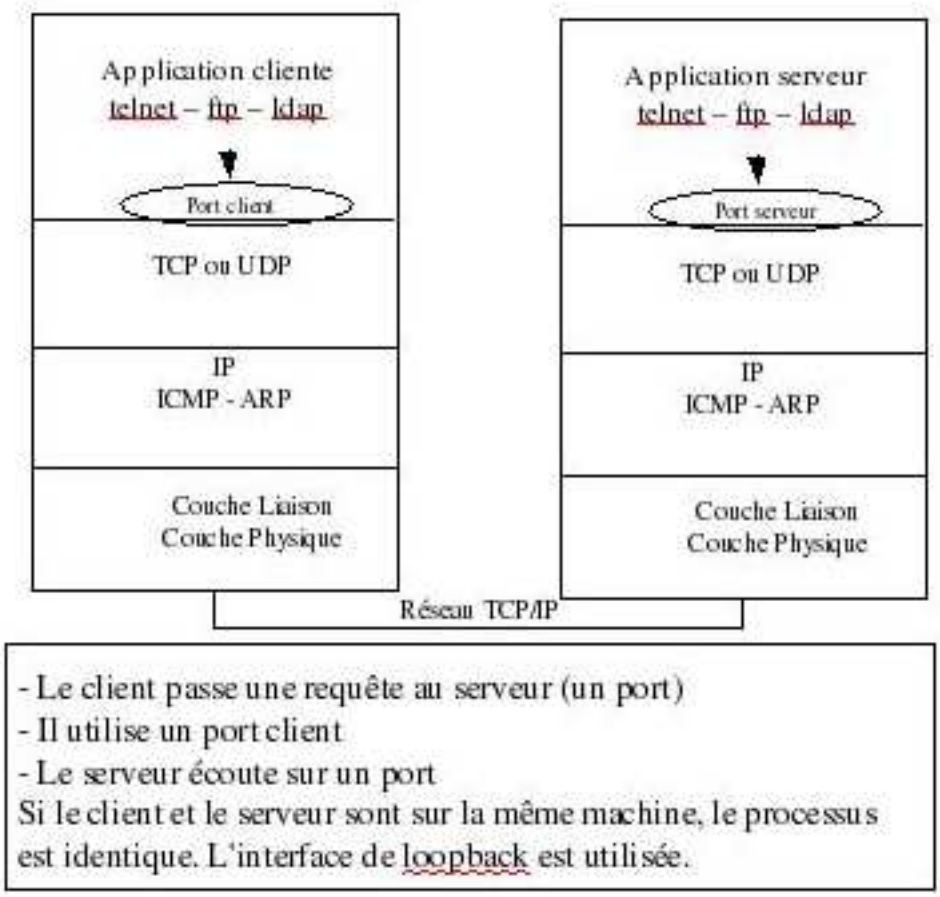
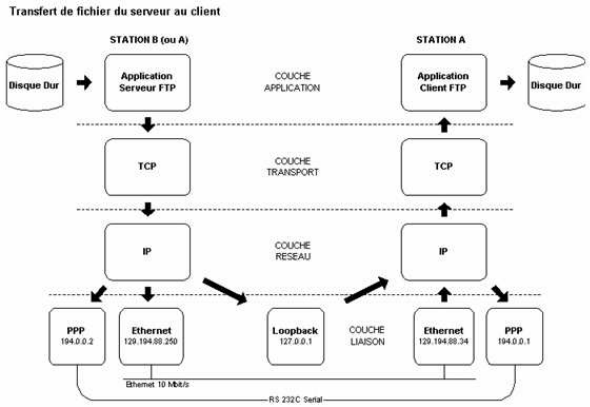


Figure 1-5. Modèle client/serveur



1.4.2. L’adressage des applicatifs : les ports

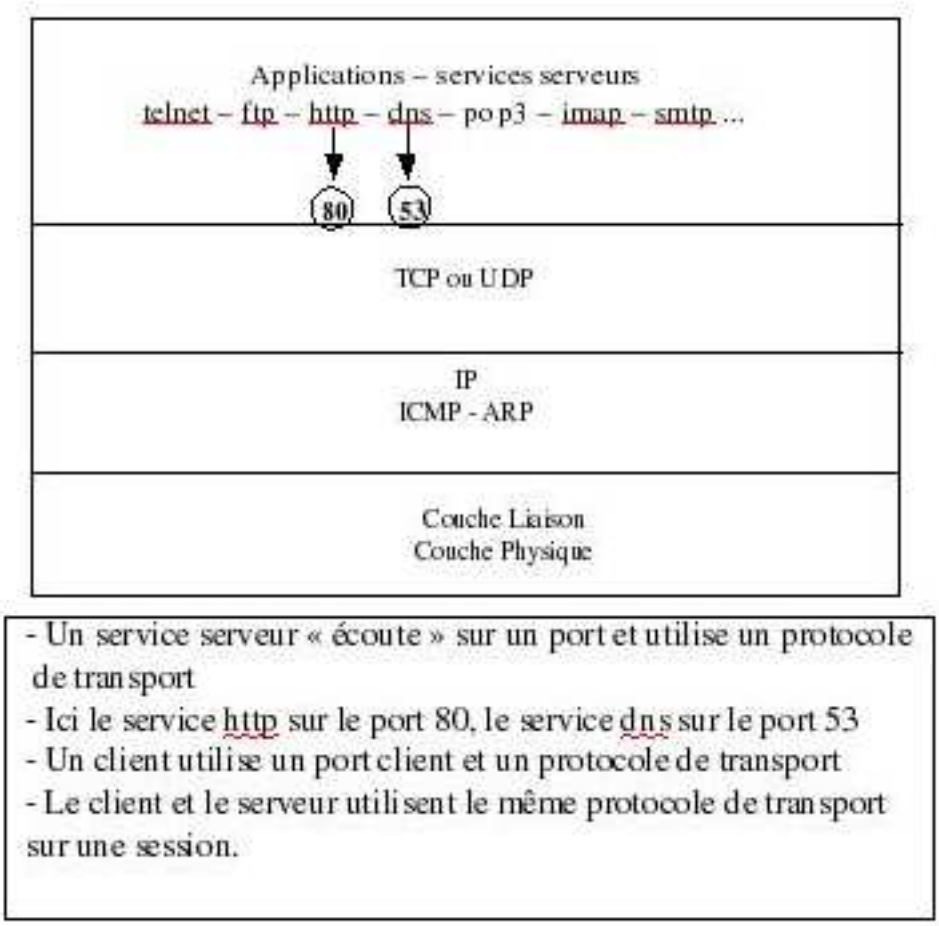
Une fois le datagramme transmis à l’hôte destinataire, il doit parvenir à l’utilisateur (si le système est multi-utilisateur) et à l’application visée (si le système est multi-tâches).

- sur la machine cliente, l’utilisateur (usager ou programme) effectue une requête vers une machine IP serveur sur le réseau. (par exemple *telnet host* ou *ftp host*). Cela se traduit par la réservation d’un port de sortie TCP ou UDP et l’envoi d’un paquet IP à la machine serveur. Ce paquet contient un message TCP ou UDP avec un numéro de port correspondant à l’application demandée sur le serveur.
- sur le serveur, la requête est réceptionnée par le pilote IP, aiguillée vers TCP ou UDP puis vers le port demandé. Le processus serveur correspondant est à l’écoute des appels sur ce port (par ex: le daemon *telnetd* traite les requêtes *telnet*, le daemon *ftpd* traite les requêtes *ftp*).
- processus client et processus serveur échangent ensuite des messages.

Des numéros de port (entre 0 et 1023) sont réservés pour les applications « standards : les ports « bien connus » (Well Known Ports), ils ont été assignés par l’IANA. Sur la plupart des systèmes ils peuvent être seulement employés par des processus du système (ou root) ou par des programmes exécutés par les utilisateurs privilégiés (liste complète : http://www.iana.org/assignments/port-numbers ou dans le fichier **/etc/services** y compris sous Windows).

D’autres numéros de port sont disponibles pour les applications développées par les utilisateurs (1024 à 65535).

Figure 1-6. Ports applicatifs



*On identifie le protocole de communication entre applications par un **numéro de protocole** et l'application par un **numéro de port**.*

Par exemple, les serveurs HTTP dialoguent de manière traditionnelle par le port 80 :

http ://www.sncf.com/index.htm <=> http :// www.sncf.com:80/index.htm

Les numéros de protocole et de port sont inclus dans le datagramme.

Une fois la connexion établie entre le client et le serveur, ceux-ci peuvent s'échanger des informations selon un protocole défini selon l'applicatif. Le client soumet des requêtes auxquelles répondra le serveur.

Ce mode de communication s'appuie sur la couche "socket". Cette couche est une interface entre la couche présentation et transport. Elle permet la mise en place du canal de communication entre le client et le serveur.

On peut schématiquement dire qu'un socket fourni un ensemble de fonctions. Ces fonctions permettent à une application client/serveur d'établir un canal de communication entre 2 ou plusieurs machines, qui utilisent un protocole de transport (TCP ou UDP) et un port de communication.

1.4.2.1. Les ports prédéfinis à connaître

Service réseau	*N° de Port*	*Type*	*Commentaire*
ICMP	7	TCP/UDP	Commandes Ping
Netstat	15	TCP/UDP	Etat du réseau
FTP	21	TCP	Transfert de fichiers
Telnet	23	TCP	Connexion de terminal réseau
SMTP	25	TCP	Envoi de courrier
DNS	53	TCP/UDP	Serveurs de noms de domaine
HTTP	80	TCP	Serveur Web
Pop3	110	TCP	Réception de courrier
sftp	115	TCP	Transfert de fichiers sécurisé
nntp	119	TCP	Service de news
ntp	123	UDP	Protocole temps réseau
nbname	137	TCP/UDP	Service de Nom Netbios
imap	143	TCP/UDP	Protocole d'accès messagerie Internet
SNMP	161	UDP	Gestion de réseau

Chapitre 2. Eléments de cours sur l’adressage IP

Le document présente l’adressage IP sur un réseau local et en environnement routé

Ce document sert d’introduction à l’ensemble des cours et TP sur les différents protocoles

Mots clés : Adressage physique, Adresse IP, masque, sous-réseau, routage

2.1. Adresses physiques (MAC) et adresses logiques (IP)

2.1.1. Notion d’adresse Physique et de trames

Deux cartes réseaux qui communiquent s’échangent des messages (suite de bits) appelés trames (frame). Tous les postes connectés au même câble reçoivent le message, mais seul celui à qui il est destiné le lit.

Comment sait-il que cette trame lui est adressée ?

Car il reconnaît l’adresse de destination, contenue dans la trame comme étant la sienne.

Comment sait il qui lui a envoyé la trame ?

Car la trame contient aussi l’adresse de l’émetteur.

Au niveau de la couche liaison, les noeuds utilisent une adresse dite « physique » pour communiquer. L’adresse correspond à l’adresse de la carte réseau. On parle **d’adresse physique**, **d’adresse MAC** (Medium Access Control) ou d’adresse de couche 2 (référence au modèle OSI).

Cette adresse est identique pour les réseaux Ethernet, Token Ring et FDDI. **Sa longueur est de 48 bits** soit six octets (par exemple : 08-00-14-57-69-69) définie par le constructeur de la carte. Une adresse universelle sur 3 octets est attribuée par l’IEEE à chaque constructeur de matériel réseau. Sur les réseaux CCITT X.25, c’est la norme X.121 qui est utilisée pour les adresses physiques, qui consistent en un nombre de 14 chiffres.

L'adresse MAC identifie de manière unique un noeud dans le monde. Elle est physiquement liée au matériel (écrite sur la PROM), c'est à dire à la carte réseau.

2.1.2. Notion d'adresse logique et de paquets

L'adresse d'une carte réseau correspond à l'adresse d'un poste et d'un seul. Or les postes sont généralement regroupés en réseau.

Comment identifier le réseau auquel appartient le poste ?

Il faut une adresse logique qui soit indépendante de l'adresse physique.

C'est ce que propose le protocole IP et le protocole IPX.

Pourquoi identifier le réseau ?

Pour permettre à 2 postes qui ne sont pas connectés au même réseau de communiquer.

Cela est impossible avec une adresse MAC, il faut une adresse de niveau supérieur, comme nous le verrons un peu plus loin et surtout avec le routage IP.

Le message véhiculé par la trame va contenir une autre adresse destinataire dont un des objectifs sera de définir le réseau destinataire du message. On appelle le message contenu dans une trame un **paquet**.

Ce qu'il nous faut savoir à ce stade, c'est qu'une machine sait que le paquet n'est pas destiné au réseau si l'adresse réseau de destination est différente de la sienne, dans ce cas elle envoie le paquet à une machine spéciale (la passerelle ou routeur) dont le rôle est d'acheminer les paquets qui sortent du réseau.

Cette adresse dite **logique** du noeud (car elle est attribuée par logiciel à un **hôte**, plus précisément à une carte réseau) contenue dans le paquet est l'adresse IP, est définie indépendamment de toute topologie d'ordinateur ou de réseau. Son format reste identique quel que soit le support utilisé.

Les machines (hôtes) d'un réseau TCP/IP sont identifiées par leur adresse IP.

3 - Résolution d'adresses logiques en adresses physiques

Toute machine sur un réseau IP a donc 2 adresses, une adresse MAC et une adresse IP.

Les processus de niveaux supérieurs utilisent toujours l'adresse IP et donc lorsqu'un processus communique avec un autre processus, il lui envoie un message dont l'adresse destinataire est une adresse IP, mais pour pouvoir atteindre la carte réseau du destinataire, il faut connaître son adresse MAC. Le rôle du protocole ARP (Adress Resolution Protocol) est d'assurer la correspondance entre l'adresse IP et l'adresse MAC.

2.1.3. Attribution d'une adresse IP Internet

Les réseaux connectés au réseau Internet mondial doivent obtenir un identificateur de réseau officiel auprès du bureau de l'Icann de l'Inter-NIC (Network Information Center) afin que soit garantie **l'unicité** des identificateurs de réseau IP sur toute la planète. Une adresse est attribuée au réseau privé dont l'administrateur en fait la demande auprès du NIC (http://www.nic.fr).

Après réception de l'identificateur de réseau, l'administrateur de réseau local doit attribuer des identificateurs d'hôte uniques aux ordinateurs connectés au réseau local. Les réseaux privés qui ne sont pas connectés à Internet peuvent parfaitement utiliser leur propre identificateur de réseau. Toutefois, l'obtention d'un identificateur de réseau valide de la part du centre InterNIC leur permet de se connecter ultérieurement à Internet sans avoir à changer les adresses des équipements en place.

Chaque noeud (interface réseau) **relié à l'Internet doit posséder une adresse IP unique**.

2.2. Adressage IP

2.2.1. Structure des adresses IP

Les **adresses IP** sont des **nombres de 32 bits** qui contiennent 2 champs :

- Un **identificateur de réseau** (NET-ID): tous les systèmes du même réseau physique doivent posséder le même identificateur de réseau, lequel doit être unique sur l'ensemble des réseaux gérés.
- Un **identificateur d'hôte** (HOST-ID): un noeud sur un réseau TCP/IP est appelé hôte, *il identifie une station de travail, un serveur, un routeur ou tout autre périphérique TCP/IP au sein du réseau.*

La concaténation de ces deux champs constitue une **adresse IP unique** sur le réseau.

Pour éviter d'avoir à manipuler des nombres binaires trop longs, les adresses 32 bits sont divisées en 4 octets. Ce format est appelé la **notation décimale pointée,** cette notation consiste à découper une adresse en quatre blocs de huit bits. Chaque bloc est ensuite converti en un nombre décimal.

Chacun des octets peut être représenté par un nombre de 0 à 255.

Ex : 130.150.0.1

Exemple :

L'adresse IP 10010110110010000000101000000001 est d'abord découpée en quatre blocs :

10010110.11001000.00001010.00000001 puis, chaque bloc est converti en un nombre décimal pour obtenir finalement 150.200.10.1

= >4 nombres entiers (entre 0 et 255) séparés par des points.

= >4 octets

L'écriture avec les points est une convention, le codage en machine est binaire.

2.2.2. Classes d'adresses

La communauté Internet a défini **trois classes d'adresses** appropriées à des réseaux de différentes tailles. Il y a, a priori, peu de réseaux de grande taille (classe A), il y a plus de réseaux de taille moyenne (classe B) et beaucoup de réseaux de petite taille (classe C). La taille du réseau est exprimée en nombre d'hôtes potentiellement connectés.

Le premier octet d'une adresse IP permet de déterminer la classe de cette adresse.

Les adresses disponibles (de 0.0.0.0 à 255.255.255.255) ont donc été découpées en plages réservées à plusieurs catégories de réseaux.

Pour éviter d'avoir recours aux organismes NIC à chaque connexion d'un nouveau poste, chaque société se voit attribuer une plage d'adresse pour son réseau. Le nombre d'adresses disponibles dans chaque plage dépend de la taille du réseau de la société. Les grands réseaux sont dits de classe A (IBM, Xerox , DEC, Hewlett-Packard), les réseaux de taille moyenne sont de classe B (Microsoft en fait partie !), et les autres sont de classe C.

Figure 2-1. Classes d'adresses

Classe	Début en binaire	Valeurs	Identificateur de réseau	Identificateur d'hôte
A	0...	*1 à 126*	*a*	*b,c,d*
B	10...	*128 à 191*	*a,b*	*c,d*
C	110...	*192 à 223*	*a,b,c*	*d*
D	1110...	*224 à 239*	multicast	*a,b,c,d*
E	1111...	*240 à 255*	réservées	expérimental

Par exemple, l'adresse d'un poste appartenant à un réseau de classe A est donc de la forme :

0AAAAAAA.xxxxxxxx.xxxxxxxx.xxxxxxxx, avec A fixé par le NIC et x quelconque.

Exemple

IBM a obtenu l'adresse 9 (en fait, on devrait dire 9.X.X.X, mais il est plus rapide de n'utiliser que la valeur du premier octet). 9 est bien de classe A car 9d=**0**0001001b

Cela signifie que chaque adresse IP du type 00001001.xxxxxxxx.xxxxxxxx.xxxxxxxx, avec x prenant la valeur 0 ou 1, fait partie du réseau d'IBM.

Malgré ces possibilités d'adressage, la capacité initialement prévue est insuffisante et sera mise à défaut d'ici quelques années. L'**IPNG** (*Internet Protocol Next Generation*) ou Ipv6 devrait permettre de résoudre ces difficultés en utilisant un adressage sur 16 octets noté en héxadécimal.

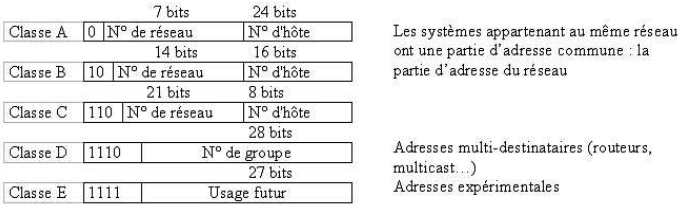
2.2.3. Identification du réseau

L'adresse IP se décompose, comme vu précédemment, en un numéro de réseau et un numéro de noeud au sein du réseau.

Afin de s'adapter aux différents besoins des utilisateurs, la taille de ces 2 champs peut varier.

On définit ainsi les 5 classes d'adresses notées A à E:

Figure 2-2. Classes d'adresses



ex. : Soit l'adresse IP suivante : 142.62.149.4

142 en décimal = 100011102 en binaire

Le mot binaire commence par les bits 102 donc il s'agit d'une adresse de classe B. Ou, plus simple : 142 est compris entre 128 et 191.

S'agissant d'une adresse de classe B, les deux premiers octets (a et b) identifient le réseau. Le numéro de réseau est donc : 142.62.0.0

Les deux derniers octets (c et d) identifient l'équipement hôte sur le réseau.

Finalement, cette adresse désigne l'équipement numéro 149.4 sur le réseau 142.62.

2.2.4. Adresses réservées

Les adresses réservées ne peuvent désigner une machine TCP/IP sur un réseau.

L'adresse d'acheminement par défaut (route par défaut.) est de type **0.X.X.X**. Tous les paquets destinés à un réseau non connu, seront dirigés vers l'interface désignée par **0.0.0.0**.

NB : 0.0.0.0 est également l'adresse utilisée par une machine pour connaître son adresse IP durant une procédure d'initialisation (DHCP).

L'adresse de bouclage (*loopback*): l'adresse de réseau 127 n'est pas attribuée à une société, elle est utilisée comme adresse de bouclage dans tous les réseaux. Cette adresse sert à tester le fonctionnement de votre carte réseau. Un ping 127.0.0.1 doit retourner un message correct. Le paquet envoyé avec cette adresse revient à l'émetteur.

Toutes les adresses de type 127.X.X.X ne peuvent pas être utilisées pour des hôtes. La valeur de 'x' est indifférente. On utilise généralement **127.0.0.1**

L'adresse de réseau est une adresse dont tous les bits d'hôte sont positionnés à 0 (ex 128.10.0.0 adresse de réseau du réseau 128.10 de classe B). Elle est utilisée pour désigner tous les postes du réseau. On utilise cette adresse dans les tables de routage.

Les noms de réseaux de type :

1. X.Y.Z.0 (de 192.0.0.0 à 223.255.255.0) sont dits de classe C

- X.Y.0.0 (de 128.0.0.0 à 191.255.0.0) sont dits de classe B :
- X.0.0.0. (de 1.0.0.0 à 126.255.255.254) sont dits de classe A :

1. de 224.0.0.0 à 254.0.0.0 : adresses réservées pour des besoins futurs
2.

L'adresse de diffusion est *une adresse dont tous les bits d'hôte sont positionnés à 1 (ex : 128.10.255.255 adresse de diffusion du réseau 128 de classe B).*

Elle est utilisée pour envoyer un message à tous les postes du réseau.

Les adresses "privées"

Les adresses suivantes (RFC 1918) peuvent également être librement utilisées pour **monter un réseau privé** :

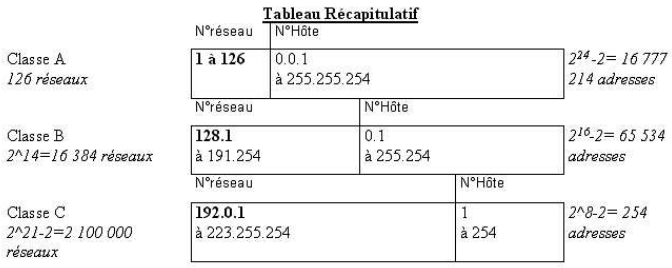
A 10.0.0.0

B 172.16.0.0 à 172.31.255.255

C 192.168.0.0 à 192.168.255.255

Aucun paquet provenant de ces réseaux ou à destination de ces réseaux, ne sera routé sur l'Internet.

Figure 2-3. Récapitulatif Classes d'adresses



Le rôle du masque de réseau (netmask) est d'identifier précisément les bits qui concernent le N° de réseau d'une adresse (il "masque" la partie hôte de l'adresse).

Un bit à 1 dans le masque précise que le bit correspondant dans l'adresse IP fait partie du **N° de réseau ;** à l'inverse, **un bit à 0** spécifie un bit utilisé pour coder le **N° d'hôte**.

Ainsi, on a un masque dit "par défaut" qui correspond à la classe de ce réseau.

Exemple: dans un réseau de classe A sans sous-réseau, le premier octet correspond à l'adresse du réseau donc le **netmask** commence par 11111111 suivi de zéros soit **255.0.0.0.**

D'où le tableau suivant :

Classe	*Netmask*
A	*255.0.0.0*
B	*255.255.0.0*
C	*255.255.255.0*

Ex : Si mon adresse IP est 149.127.1.110 alors je travaille avec une adresse de classe B. Mon N° de réseau est 149.127.0.0 et mon masque 255.255.0.0.

2.3. Les sous-réseaux

2.3.1. Pourquoi créer des sous réseaux ?

Les avantages de la segmentation en sous-réseau sont les suivants :

1. **Utilisation de plusieurs media** (câbles, supports physiques). La connexion de tous les noeuds à un seul support de réseau peut s'avérer impossible, difficile ou coûteuse lorsque les noeuds sont trop éloignés les uns des autres ou qu'ils sont déjà connectés à un autre media.
2. **Réduction de l'encombrement**. Le trafic entre les noeuds répartis sur un réseau unique utilise la largeur de bande du réseau. Par conséquent, plus les noeuds sont nombreux, plus la largeur de bande requise est importante. La répartition des noeuds sur des réseaux séparés permet de réduire le nombre de noeuds par réseau. Si les noeuds d'un réseau de petite taille communiquent principalement avec d'autres noeuds du même réseau, l'encombrement global est réduit.
3. **Economise les temps de calcul**. Les diffusions (paquet adressé à tous) sur un réseau obligent chacun des noeuds du réseau à réagir avant de l'accepter ou de la rejeter.
4. **Isolation d'un réseau**. La division d'un grand réseau en plusieurs réseaux de taille inférieure permet de limiter l'impact d'éventuelles défaillances sur le réseau concerné. Il peut s'agir d'une erreur matérielle du réseau (une connexion
5. **Renforcement de la sécurité**. Sur un support de diffusion du réseau comme Ethernet, tous les noeuds ont accès aux paquets envoyés sur ce réseau. Si le trafic sensible n'est autorisé que sur un réseau, les autres hôtes du réseau n'y ont pas accès.
6. **Optimisation de l'espace réservé à une adresse IP**. Si un numéro de réseau de classe A ou B vous est assigné et que vous disposez de plusieurs petits réseaux physiques, vous pouvez répartir l'espace de l'adresse IP en multiples sous-réseaux IP et les assigner à des réseaux physiques spécifiques. Cette méthode permet d'éviter l'utilisation de numéros de réseau IP supplémentaires pour chaque réseau physique.

2.3.2. Masque de sous-réseau

Les masques de sous-réseaux (*subnet mask*) permettent de **segmenter un réseau en plusieurs sous-réseaux**. On utilise alors une partie des bits de l'adresse d'hôte pour identifier des sous-réseaux.

L'adressage de sous-réseau permet de définir des organisations internes de réseaux qui ne sont pas visibles à l'extérieur de l'organisation. Cet adressage permet par exemple l'utilisation d'un routeur externe qui fournit alors une seule connexion Internet.

Toutes les machines appartenant à un sous-réseau possèdent le même numéro de réseau.

On utilise le même principe que pour le masque par défaut sur l'octet de la partie hôte auquel on va prendre des bits. Ainsi, le masque de sous-réseau d'une adresse de classe B commencera toujours par 255.255.xx.xx

Pour connaître l'adresse du sous-réseau auquel une machine appartient, on effectue en réalité un **ET logique** entre l'adresse de la machine et le masque.

Adresse : 200.100.40.33 11001000.01100100.00101000.00100001

Masque : 255.255.255.224 11111111.11111111.11111111.11100000

Opération ET 11001000.01100100.00101000.00100000

=> La machine appartient au sous-réseau : 200.100.40.32

Nous voyons dans ce deuxième exemple que nous avons pris 3 bits sur le dernier octet de notre adresse. Ces 3 bits vont nous permettre de construire plusieurs sous-réseaux.

Ex : adresse : 192.0.0.131

Masque : 255.255.255.192

Conversion de l'adresse en binaire : *11000000 00000000 00000000 10000011*

Conversion du masque en binaire : **11111111 11111111 11111111 11000000**

La machine appartient au sous-réseau *192.0.0.192 et a l'adresse 11=3*

Pour des raisons de commodité, on préférera **réserver un octet entier** pour coder le numéro de sous réseau. De même la théorie ne nous oblige pas à prendre les **bits contigus d'un masque**, même si c'est ce que nous utiliserons en pratique.

Important : pour parer à d'éventuels problèmes de routage et d'adressage, tous les ordinateurs d'un réseau logique doivent utiliser le même masque de sous-réseau et le même identificateur de réseau.

2.3.3. Sous-réseaux

2.3.3.1. Nombre de sous-réseaux

Le nombre théorique de sous-réseaux est égal à 2n, n étant le nombre de bits à 1 du masque, utilisés pour coder les sous-réseaux.

Exemple :

Adresse de réseau : 200.100.40.0

Masque : 255.255.255.224

224 = 11100000 donc **3 bits** pour le N° de **sous-réseau** et 5 bits pour l'hôte.

Le nombre de sous-réseau est donc de *: 23 =8.*

Remarque : la RFC 1860 (remplacée par la RFC 1878) stipulait qu'un numéro de sous réseau ne peut être composé de bits tous positionnés à zéro ou tous positionnés à un.

Autrement dit, dans notre exemple, on ne pouvait pas utiliser le sous-réseau 0 et le sous-réseau 224. Le premier nous donnant une adresse de sous-réseau équivalente à l'adresse du réseau soit 200.100.40.0. Le deuxième nous donnant une adresse de sous-réseau dont l'adresse de diffusion se confondrait avec l'adresse de diffusion du réseau. Le nombre de sous-réseaux aurait alors été de seulement *: 2^3-2 =6.*

Il est donc important de savoir quelle RFC est utilisée par votre matériel pour savoir si les adresses de sous-réseau composées de bits tous positionnés à zéro ou tous positionnés à un sont prises en compte ou non.

2.3.3.2. Adresse des sous-réseaux

Il faut donc maintenant trouver les adresses des sous-réseaux valides en utilisant les bits à 1 du masque.

Pour l'exemple précédent, il faut utiliser les 3 premiers bits:

000 00000 = 0

001 00000 = 32

010 00000 = 64

011 00000 = 96

100 00000 = 128

101 00000 = 160

110 00000 = 192

111 00000 = 224

On constate que le pas entre 2 adresses de sous-réseau est de *32 = 25* correspondant *au nombre théorique d'hôtes par sous-réseau.*

2.3.3.3. Adresse de diffusion d'un sous-réseau

Il faut mettre **tous les bits de la partie hôte à 1**.

Cherchons l'adresse de diffusion des sous réseaux précédents.

- Avec le masque 255.255.255.224

Pour le sous-réseau 200.100.40.**32**

32 = **001** 00000 donc l'adresse de diffusion est *001 11111 = 63.*

L'adresse de diffusion complète est donc *200.100.40.63*

Pour le sous-réseau 200.100.40.**64** l'adresse de diffusion est 200.100.40.*95*

...ETC ...

Avec le masque 255.255.255.129

Pour le sous-réseau 200.100.40.1 l'adresse de diffusion est 200.100.40.127

Pour le sous-réseau 200.100.40.128 l'adresse de diffusion est 200.100.40.254

Pourquoi 254 et pas 255 car avec 255 le dernier bit serait à 1 donc on serait dans le sous-réseau 10000001 , en décimal 129.

2.3.3.4. Nombre de postes d'un sous-réseau

Le nombre de postes est égal à **2n**, **n** étant le nombre de **bits à 0** du masque permettant de coder l'hôte. A ce chiffre il faut enlever 2 numéros réservés :

- tous les bits à zéro qui identifie le sous-réseau lui-même.
- tous les bits à 1 qui est l'adresse de diffusion pour le sous-réseau.

Exemples :

Soit le masque 255.255.255.224

224 = 11100000 donc 3 bits pour le N° de sous-réseau et 5 bits pour l'hôte

le nombre de poste est donc de : 2^5 -2 =30 postes.

De même, avec le masque 255.255.255.129 le nombre de postes sera de *2^6-2 = 62 postes*

2.3.3.5. Adresse de poste sur un sous-réseau

L'adresse de poste sur un sous-réseau subnetté " normalement " ne pose pas de problème, elle est comprise dans la fourchette [adresse de sous-réseau + 1, adresse de diffusion du sous-réseau - 1] soit dans l'exemple précédent :

[200.100.400.33,200.100.40.62] pour le sous-réseau 200.100.40.32

[200.100.400.65,200.100.40.94] pour le sous-réseau 200.100.40.64.

Par exemple, au lieu d'allouer un identificateur de réseau de classe B, dans une entreprise comportant 2000 hôtes, InterNic alloue une plage séquentielle de 8 identificateurs de réseau de classe C. Chaque identificateur de réseau de classe C gère 254 hôtes pour un total de 2 032 identificateurs d'hôte.

Alors que cette technique permet de conserver des identificateurs de réseau de classe B, elle crée un nouveau problème.

En utilisant des techniques de routage conventionnelles, les routeurs d'lnternet doivent désormais comporter huit entrées (en RAM) dans leurs tables de routage pour acheminer les paquets IP vers l'entreprise. La technique appelée CIDR (Classless Inter-Domain Routing) permet de réduire les huit entrées utilisées dans l'exemple précédent à une seule entrée correspondant à tous les identificateurs de réseau de classe C utilisés par cette entreprise.

Soit les huit identificateurs de réseau de classe C commençant par l'identificateur de réseau 220.78.**168.0** et se terminant par l'identificateur de réseau 220.78.**175.0**, l'entrée de la table de routage des routeurs d'lnternet devient :

Identificateur de réseau	*Masque de sous réseau*	*Masque de sous réseau (en binaire)*
220.78.168.0	*255.255.248.0*	11111111 11111111 *11111*000 00000000

En effet 168 en binaire donne : **10101**000

et 175 donne : **10101**111

la partie commune porte bien sur les 5 1ers bits

d'où le masque : **11111**000

Dans l'adressage de sur-réseaux, la destination d'un paquet est déterminée en faisant un ET logique entre l'adresse IP de destination et le masque de sous-réseau de l'entrée de routage. En cas de correspondance avec l'identificateur de réseau, la route est utilisée. Cette procédure est identique à celle définie pour l'adressage de sous-réseaux.

La notation CIDR définit une convention d'écriture qui spécifie le nombre de bits utilisés pour identifier la partie réseau (les bits à 1 du masque).

Les adresses IP sont alors données sous la forme :

142.12.42.145 / **24** <=> 142.12.42.145 255.255.255.0

153.121.219.14 / **20**<=> 153.121.219.14 255.255.240.0

Dans cette écriture les nombres **24** et **20** représentent le nombre de bits consacrés à la codification du réseau (et sous réseau).

Remarque : Les RFC 1518 et 1519 définissent le CIDR *(Classless Inter-Domain Routing).*

2.4. Le routage

2.4.1. Recherche de l'adresse physique

La communication entre machines ne peut avoir lieu que lorsque celles-ci connaissent leurs adresses physiques (MAC). Pour envoyer les paquets IP vers les autres noeuds du réseau, les noeuds qui utilisent les protocoles TCP/IP **traduisent les adresses IP de destination en adresses MAC**. L'application émettrice ajoute son adresse IP au paquet et l'application réceptrice peut utiliser cette adresse IP pour répondre.

Sur les réseaux à diffusion, tels qu'Ethernet et Token-Ring, le **protocole** IP nommé **ARP** (*Address Resolution Protocol*) fait le lien entre les adresses IP et les adresses physiques (ou MAC).

Quand un poste **cherche l'adresse physique** correspondant à l'adresse IP qu'il connaît, le protocole **ARP** se met en oeuvre et réalise les tâches suivantes :

1. réalisation d'un appel broadcast sur le réseau en demandant à qui correspond l'adresse IP à résoudre : il diffuse un paquet ARP qui contient l'adresse IP du destinataire
2. les machines du réseau comparent l'adresse demandée à leur adresse et le noeud correspondant renvoie son adresse physique au noeud qui a émis la requête.
3. stockage de l'adresse physique lorsque le destinataire répond dans le cache ARP de la machine

Pour accélérer la transmission des paquets et réduire le nombre des requêtes de diffusion qui doivent être examinées par tous les noeuds du réseau, chaque noeud dispose d'un **cache de résolution d'adresse**. Chaque fois que le noeud diffuse une requête ARP et reçoit une réponse, il crée une entrée dans une table de correspondance stockée en mémoire cache. Cette entrée assigne l'adresse IP à l'adresse physique.

Lorsque le noeud envoie un autre paquet IP, il cherche l'adresse IP dans son cache. S'il la trouve, il utilise alors l'adresse physique correspondante pour son paquet.

Le noeud diffuse une requête ARP seulement s'il ne trouve pas l'adresse IP dans son cache.

2.4.2. Principe

Le routage dans Internet est similaire au **mécanisme d'adressage du courrier**.

Si vous adressez une lettre à un destinataire aux USA, à Los Angeles, dans l'état de Californie. Le bureau de poste de Belfort reconnaîtra que cette adresse n'est pas locale et transmettra le courrier au bureau français des PTT qui le remettra au service du mail US. Celui-ci s'en remettra à son bureau de la Californie, qui le transmettra au bureau de Los Angeles, qui connaît la localisation qui correspond à l'adresse dans la ville.

Avantages du système :

1. le bureau de poste local n'a pas à connaître toutes les adresses du monde
2. le chemin suivi peut être variable : chaque opérateur sait juste à qui remettre le courrier.

Le routage dans un réseau est identique :

Internet en entier est composé de réseaux autonomes qui s'occupent en interne de l'adressage entre leurs hôtes. Ainsi, tout datagramme arrivant sur un hôte quelconque du réseau destination sera acheminé à bon

port par ce réseau seul.

Quand tous les hôtes participent au même réseau, chacun d'eux peut adresser des paquets aux autres sans difficulté. Par contre, si le destinataire est situé sur un autre réseau, le problème est de savoir où et à qui adresser le paquet puisque l'hôte expéditeur ne « voit » pas le destinataire.

On appelle **passerelle** (dans la terminologie TCP/IP) ou **routeur** un équipement qui fait le lien entre différents réseaux ou entre sous-réseaux. Ex de passerelle: **un ordinateur équipé de plusieurs adaptateurs réseau** peut être relié avec chacune d'elle à un réseau physiquement séparé.

Les paquets d'un réseau qui sont adressés à l'autre réseau doivent passer par la passerelle. D'où la nécessité pour chaque hôte de connaître, sur son réseau, l'adresse IP d'un ou de plusieurs routeurs qui servent de passage vers le ou les réseaux qu'ils ne connaît pas.

Mettre en place le routage consiste à configurer chaque hôte du réseau de façon à ce qu'il sache vers quelle adresse de son propre réseau il doit adresser un paquet qui concerne un autre réseau (ou sous-réseau). Ces destinataires intermédiaires sont des routeurs qui prennent en charge le paquet.

Les hôtes pouvant être nombreux, bien souvent chacun ne connaît que l'adresse d'une passerelle (routeur) par défaut et ce sera cette passerelle qui « connaîtra » les adresses des autres routeurs.

2.4.3. Acheminement des paquets TCP-IP

Voici comment un hôte expéditeur se comporte pour adresser un paquet à un destinataire :

1. Il extrait l'adresse de réseau, voire de sous réseau de l'adresse du destinataire et la compare à sa propre adresse de réseau ou de sous réseau. S'il s'agit du même réseau, le paquet est expédié directement au destinataire en mettant en oeuvre ARP.
2. S'il ne s'agit pas du même réseau, l'expéditeur cherche dans sa table de routage une correspondance destinataire final / destinataire intermédiaire (routeur). Il cherche, en quelque sorte, sur son réseau, un hôte capable de servir de facteur vers un autre réseau.
3. L'expéditeur cherche d'abord à trouver dans sa table de routage locale l'adresse IP complète du destinataire,
4. s'il ne la trouve pas il cherche l'adresse du sous réseau du destinataire,
5. s'il ne la trouve pas, il cherche enfin l'adresse du réseau,
6. s'il ne trouve aucune correspondance, l'expéditeur cherche dans sa table l'adresse d'une passerelle à utiliser par défaut, (route 0.0.0.0)
7. s'il échoue là encore, le paquet, décidément bien encombrant, est supprimé.
8. Si l'une de ces recherches aboutit, la machine émettrice construit le paquet avec l'adresse IP du destinataire hors réseau. Elle l'encapsule dans une trame ayant comme adresse MAC de destination l'adresse MAC du routeur. La couche 2 du routeur lit la trame qui lui est adressée et la transmet à la

couche 3 IP. Celle-ci récupère le paquet et s'aperçoit que le paquet ne lui est pas adressé, elle consulte sa table de routage, décide sur quelle nouvelle interface réseau le paquet doit être transmis, encapsule le paquet dans une nouvelle trame, et ainsi de suite de passerelle en passerelle jusqu'à destination.

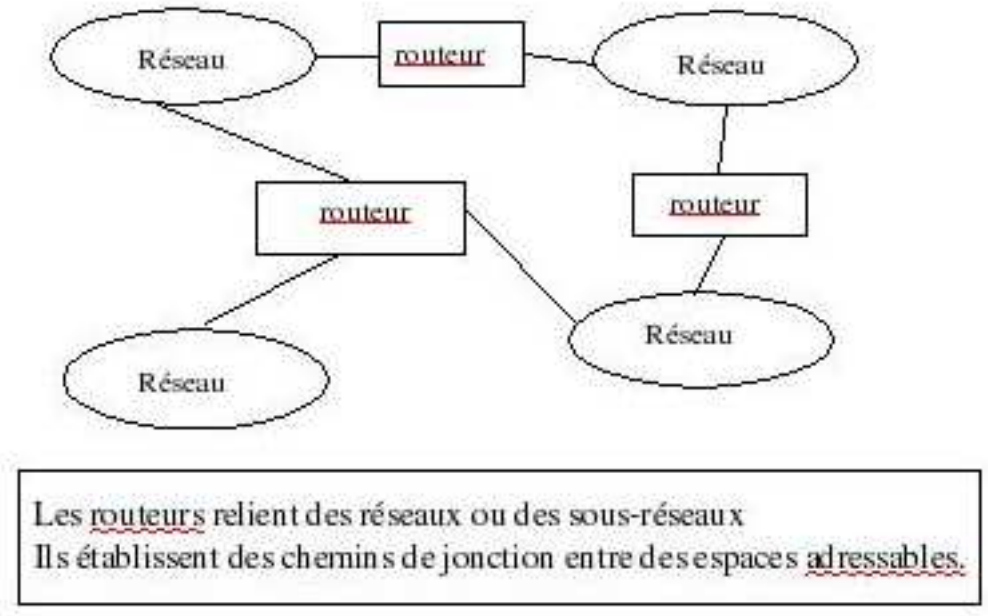
2.4.4. Les tables de routage

Les réseaux IP sont interconnectés par des routeurs IP de niveau 3 (appelés abusivement en terminologie IP des gateways ou passerelles).

Chaque station IP doit connaître le routeur par lequel il faut sortir pour pouvoir atteindre un réseau extérieur, c'est-à-dire avoir en mémoire une table des réseaux et des routeurs. Pour cela elle contient une table de routage locale.

Dans une configuration de **routage statique**, une table de correspondance entre adresses de destination et adresses de routeurs intermédiaires est complétée « à la main » par l'administrateur, on parle de **table de routage**.

Figure 2-4. table de routage



Réseau 1 --> Routeur 1

Réseau 2 --> Routeur 1

.....

Réseau n --> Routeur p

La table de routage comporte les adresses des passerelles permettant d'atteindre les réseaux de destination. La commande **Route** permet de manipuler le contenu de la table de routage.

Exemple de table de routage :

Destination	*Masque de Sous réseau*	*Passerelle*	
127.0.0.1	255.255.255.0	127.0.0.1	voie de bouclage
142.62.10.0	255.255.255.0	142.62.10.99	sortie de la passerelle vers le sous-réseau 10
142.62.20.0	255.255.255.0	142.62.20.99	sortie de la passerelle vers le sous-réseau 20

2.4.5. Acheminement Internet

2.4.5.1. Domaine d'acheminement

Les échanges entre passerelles de chaque domaine de routage font l'objet de protocoles particuliers : EGP (Exterior Gateway Protocol) et BGP (Border Gateway Protocol) plus récent. Ces protocoles envoient les paquets vers des destinations en dehors du réseau local vers des réseaux externes (Internet, Extranet...).

2.4.5.2. Principe du choix d'une voie d'acheminement

1. Si l'hôte de destination se trouve sur le réseau local, les données sont transmises à l'hôte destination
2. Si l'hôte destination se trouve sur un réseau à distance, les données sont expédiées vers une passerelle locale qui route le paquet vers une autre passerelle et ainsi de suite de passerelle en passerelle jusqu'à destination.

La commande **Tracert** permet de suivre à la trace le passage de routeur en routeur pour atteindre un hôte sur le réseau. La commande **Ping** permet de vérifier la fiabilité d'une route donnée.

2.4.6. Routage dynamique

Les **protocoles d'échange dynamique des tables de routage IP** sur un réseau local sont **RIP** (*Routing Information Protocol*) et le protocole **OSPF (Open Shortest Path First).** Dans une configuration de **routage dynamique**, un protocole (RIP ou OSPF) est mis en oeuvre pour construire dynamiquement les chemins entre routeurs.

Le protocole RIP permet à un routeur d'échanger des informations de routage avec les routeurs avoisinants. Dès qu'un routeur est informé d'une modification quelconque de la configuration sur les réseaux (telle que l'arrêt d'un routeur), il transmet ces informations aux routeurs avoisinants. Les routeurs envoient également des paquets de diffusion générale RIP périodiques contenant toutes les informations de routage dont ils disposent. Ces diffusions générales assurent la synchronisation entre tous les routeurs.

Avec un protocole comme RIP, on peut considérer que les tables de routages des routeurs et passerelles sont constituées et mises à jour automatiquement.

Chapitre 3. Eléments de cours sur ARP

Résumé sur ARP

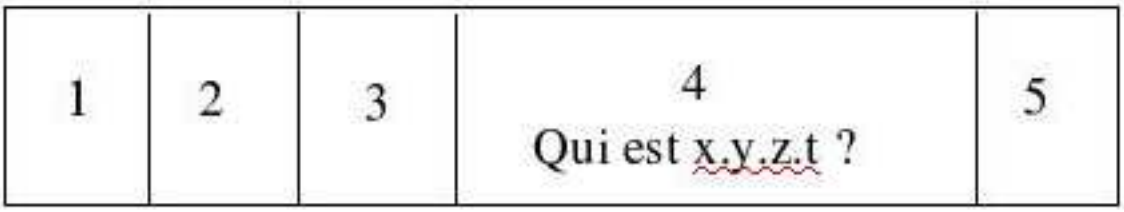
3.1. Le protocole ARP

L'adresse Ethernet est une adresse unique sur 48 bits (6 octets) associée à la carte Ethernet. Lorsqu'un noeud `N1` du réseau TCP/IP X1.X1.X1.X1 veut émettre un paquet TCP/IP (dans une trame Ethernet) vers une machine `N2` d'adresse IP (X2.X2.X2.X2), il faut qu'il connaisse l'adresse Ethernet (E2.E2.E2.E2.E2.E2). Pour réaliser l'association *@ip / @ Ethernet* l'émetteur `N1` utilise le protocole ARP dont le principe est le suivant :

L'émetteur envoie une trame Ethernet de diffusion (broadcast) (ie @destinataire toute à 1) contenant un message ARP demandant

qui est X2.X2.X2.X2 ?

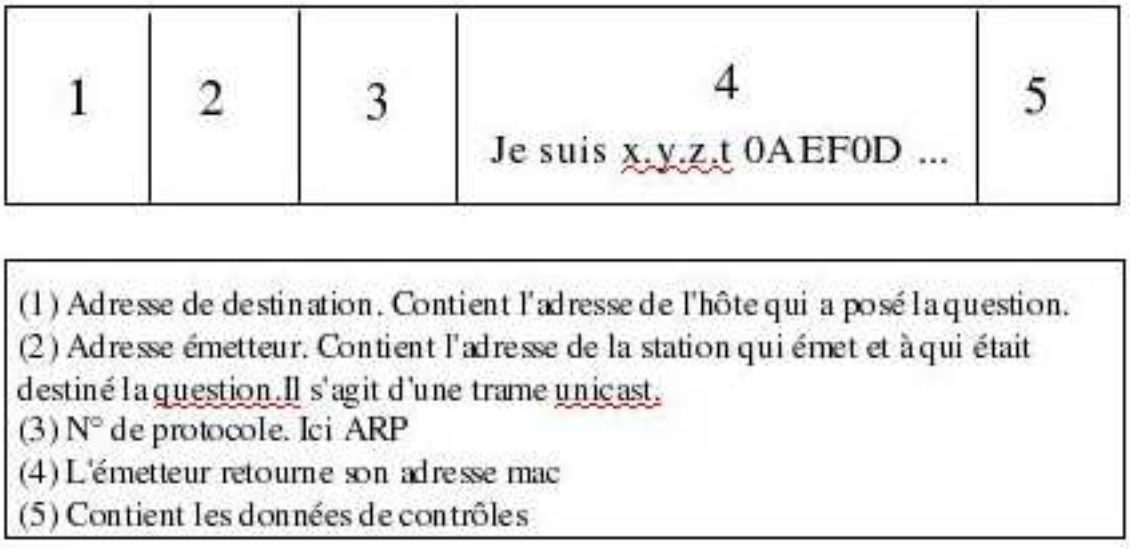
Figure 3-1. Trame Ethernet contenant une requête ARP



(1) Adresse de destination. Contient 255.255.255.255 Il s'agit d'une trame de broadcast.
(2) Adresse émetteur. Contient l'adresse de la station qui émet
(3) N° de protocole. Ici ARP
(4) Contient la question (recherche de l'adresse mac de x.y.z.t). La question est posée à tout le monde. Seule la station concernée répondra.
(5) Contient les données de contrôles

Toutes les machines IP du réseau local reçoivent la requête. `N2` qui a l'adresse X2.X2.X2.X2 se reconnaît, et elle répond à `N1` ie X1.X1.X1.X1 (dans une trame destinée à E1.E1.E1.E1.E1.E1)

Figure 3-2. Trame Ethernet contenant une réponse ARP



Chaque machine maintient en mémoire une table cachée de correspondances *@ip / @ Ethernet* pour éviter trop de requêtes ARP. Chaque entrée de la table à une durée de vie limitée. Voici pour exemple ce que donne le programme tcpdump avec la commande **ping *`192.168.1.2`*** à partir de la machine `uranus` alors que la table ARP de l'hôte `uranus` est vide :

```
13:17:14.490500 arp who-has 192.168.1.2 tell uranus.planete.net
13:17:14.490500 arp reply 192.168.1.2 is-at 0:40:33:2d:b5:dd
13:17:14.490500 uranus.planete.net > 192.168.1.2: icmp: echo request
13:17:14.490500 192.168.1.2 > uranus.planete.net: icmp: echo reply
13:17:15.500500 uranus.planete.net > 192.168.1.2: icmp: echo request
13:17:15.500500 192.168.1.2 > uranus.planete.net: icmp: echo reply
```

Explications :

Ligne 1, `uranus` demande qui est 192.168.1.2 (requête ARP) Le paquet est diffusé à tous les hôtes du réseau.

Ligne 2 réponse ARP : je suis à l'adresse Ethernet 00:40:33:2d:b5:dd

Lignes 3 à 6 : échanges de paquets ICMP entre les 2 hôtes.

Chapitre 4. L’adressage IP v6

L’adressage IPv4 sur 32 bits se révélant insuffisant (saturation prévue pour 2010) avec le développement d’Internet, l’IETF en 1998 a proposé le standard IPv6 (ou Ipng - ng pour "Next Generation", RFC 2460), afin de permettre l’adressage d’au moins un milliard de réseaux, soit quatre fois plus qu’IPv4.

IPv6 possède un nouveau format d’en-tête IP, une infrastructure de routage plus efficace, et un espace d’adressage plus important. Pour permettre le déploiement d’IPv6 de la manière la plus flexible possible, la compatibilité avec IPv4 est garantie.

4.1. Caractéristiques

- les **adresses IPv6 sont codées sur 128 bits** (1 milliard de réseaux).

- le principe des numéros de réseaux et des numéros d’hôtes est maintenu.

- IPv6 est conçu pour interopérer avec les systèmes IPv4 (transition douce prévue sur 20 ans). L’adresse IPv6 peut contenir une adresse IPv4 : on place les 32 bits de IPv4 dans les bits de poids faibles et on ajoute un préfixe de 96 bits (80 bits à 0 suivis de 16 bits à 0 ou 1)

- IPv6 utilise un **adressage hiérarchique** (identification des différents réseaux de chaque niveau) ce qui permet un routage plus efficace.

- IPv6 prévu pour les systèmes mobiles : auto-configuration, notion de voisinage (neighbor).

- IPv6 permet **l’authentification et le chiffrement** dans l’en-tête des paquets, ce qui permet de sécuriser les échanges. En effet IP v.6 intègre **IPSec** (protocole de création de tunnel IP avec chiffrement), qui garantit un contexte sécurisé par défaut.

- IPv6 intègre la **qualité de service** : introduction de flux étiquetés (avec des priorités)

- IPv6 prend mieux en charge le trafic en temps réel (garantie sur le délai maximal de transmission de datagrammes sur le réseau).

4.2. Types d’adresses

IPv6 supporte 3 types d’adresses: Unicast, Anycast et Multicast.

Anycast est un nouveau type d’adressage. Il identifie qu’un noeud, parmi un groupe de noeuds, doit recevoir l’information. L’interface de destination doit spécifiquement être configurée pour savoir qu’elle est Anycast.

La notion de diffusion (broadcast) disparaît dans IPv6.

4.3. Représentation des adresses

Une adresse IPv6 s’exprime en notation hexadécimale avec le séparateur "deux-points".

Exemple d’adresse :

5800:10C3:E3C3:F1AA:48E3:D923:D494:AAFF

Dans IPv6 les masques sont exprimés en notation CIDR.

Il y a 3 façons de représenter les adresses IPv6

forme préférée :

"x:x:x:x:x:x:x:x" où x représente les valeurs hexadécimales des 8 portions de 16 bits de l’adresse. Exemple:

3ffe:0104:0103:00a0:0a00:20ff:fe0a:3ff7

forme abrégée :

Un groupe de plusieurs champs de 16 bits mis à 0 peut être remplacé par la notation "::".

La séquence "::" ne peut apparaître qu’une seule fois dans une adresse.

Exemple:

5f06:b500:89c2:a100::800:200a:3ff7

ff80::800:200a:3ff7

::1

forme mixte :

Lorsqu’on est dans un environnement IPv4 et IPv6, il est possible d’utiliser une représentation textuelle de la forme: "**x:x:x:x:x:x:d.d.d.d**", où les ’x’ sont les valeurs hexadécimales des 6 premiers champs de 16 bits et les ’d’ sont les valeurs décimales des 4 derniers champs de 8 bits de l’adresse.

Exemple:

::137.194.168.93

4.4. Allocation de l’espace d’adressage

Le type d’adresse IPv6 est indiqué par les premiers bits de l’adresse qui sont appelés le "Préfixe de Format" (Format Prefix). L’allocation initiale de ces préfixes est la suivante:

Allocation	*Préfixe*	*Usage*
Adresses Unicast pour ISP	010	Adresse d’un hôte sur Internet
Adresses Unicast expérimentales	001	
Adresses "Link Local Use"	1111 1110 10	Un seul réseau, autoconfiguration, « neighbor »
Adresses "Site Local Use"	1111 1110 11	sous-réseaux privés
Adresses Multicast	1111 1111	

15 % de l’espace d’adressage est actuellement alloué. Les 85% restants sont réservés pour des usages futurs. En réalité sur les 128 bits, seulement 64 sont utilisés pour les hôtes (Interface ID).

Chapitre 5. Fichiers de configuration du réseau et commandes de base

Présentation des principaux fichiers de configuration du réseau et des commandes d'administration système et réseau.

5.1. Présentation du document : les outils de l'administrateur réseau

Ce document présente les principaux fichiers de configuration d'une machine en réseau, et les commandes d'administration réseau.

Il est composé de 6 parties:

1. Les fichiers de configuration réseau
2. La commande **ifconfig**
3. La commande **arp**
4. La commande **route**
5. La commande **netstat**
6. La commande **traceroute**

5.2. Les fichiers de configuration

5.2.1. Le fichier `/etc/hosts`

Le fichier `hosts` donne un moyen d'assurer la résolution de noms

Exemple de fichier `host`

```
127.0.0.1  localhost localhost.localdomain
192.168.1.1  uranus.foo.org uranus
```

5.2.2. Le fichier `/etc/networks`

Il permet d'affecter un nom logique à un réseau

```
localnet   127.0.0.0
foo-net 192.168.1.0
```

Cette option permet par exemple d'adresser un réseau sur son nom, plutôt que sur son adresse.

route add *`foo-net`* au lieu de **route add `-net` *`192.168.1.0`***.

5.2.3. Le fichier `/etc/host.conf`

Il donne l'ordre dans lequel le processus de résolution de noms est effectué. Voici un exemple de ce que l'on peut trouver dans ce fichier :

```
order hosts,bind
```

La résolution est effectuée d'abord avec le fichier `host`, en cas d'échec avec le DNS.

5.2.4. Le fichier `/etc/resolv.conf`

Il permet d'affecter les serveurs de noms.

Exemple

```
Nameserver 192.168.1.1
Nameserver 192.168.1.2
Nameserver 192.168.1.3
```

Ici le fichier déclare le nom de domaine et 3 machines chargées de la résolution de noms.

5.2.5. Les fichiers de configuration des interfaces réseau

Vous trouverez ces fichiers dans `/etc/network/interfaces`. Voici un exemple qui contient 3 interfaces.

```
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)
# The loopback interface
# automatically added when upgrading
```

```
auto lo eth0 eth1

iface lo inet loopback

iface eth0 inet static
        address 192.168.90.1
        netmask 255.255.255.0
        network 192.168.90.0
        broadcast 192.168.90.255
        gateway 192.168.90.1

iface eth1 inet static
        address 192.168.0.1
        netmask 255.255.255.0
        network 192.168.0.0
        broadcast 192.168.0.255
```

5.3. Les outils de l’administrateur réseau

5.3.1. La commande ifconfig

La commande **ifconfig** permet la configuration locale ou à distance des interfaces réseau de tous types d’équipements (unité centrale, switch, routeur). La ligne de commande est :

`ifconfig` ***`interface adresse [parametres]`***.

Exemple : **`ifconfig`** ***`eth0 192.168.1.2`*** (affecte l’adresse 192.168.1.2 à la première interface physique.

Voici les principaux arguments utilisés :

`interface` logique ou physique, il est obligatoire,

`up` active l’interface

`down` désactive l’interface

`mtu` définit l’unité de transfert des paquets

`netmask` affecter un masque de sous-réseau

`broadcast` définit l'adresse de broadcast

`arp` ou `-arp` activer ou désactiver l'utilisation du cache arp de l'interface

`metric` paramètre utilisé pour l'établissement des routes dynamiques, et déterminer le « coût » (nombre de sauts ou « hops ») d'un chemin par le protocole RIP.

`multicast` active ou non la communication avec des machines qui sont hors du réseau.

`promisc` ou `-promisc` activer ou désactiver le mode promiscuité de l'interface. En mode *promiscuous*, tous les paquets qui transitent sur le réseau sont reçus également par l'interface. Cela permet de mettre en place un analyseur de trame ou de protocole.

*Description du résultat de la commande **ifconfig `eth0`** :*

1. eth0 Link encap:Ethernet HWaddr 00:80:C8:32:C8:1E
2. inet addr:192.168.1.1 Bcast:192.168.1.255 Mask:255.255.255.0
3. UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
4. RX packets:864 errors:0 dropped:0 overruns:0 frame:0
5. TX packets:654 errors:0 dropped:0 overruns:0 carrier:0
6. collisions:0
7. Interrupt:10 Base address:0x6100

Explications :

Ligne 1: l'interface est de type Ethernet. La commande nous donne l'adresse MAC de l'interface.

Ligne 2 : on a l'adresse IP celle de broadcast, celle du masque de sous-réseau

Ligne 3 : l'interface est active (UP), les modes broadcast et multicast le sont également, le MTU est de 1500 octets, le Metric de 1

Ligne 4 et 5 : RX (paquets reçus), TX (transmis), erreurs, suppressions, engorgements, collision

Mode d'utilisation :

Ce paragraphe décrit une suite de manipulation de la commande **ifconfig**.

Ouvrez une session en mode console sur une machine.

1 - Relevez les paramètres de votre machine à l'aide de la commande **ifconfig**. Si votre machine n'a qu'une interface physique, vous devriez avoir quelque chose d'équivalent à cela.

```
Lo  Link encap:Local Loopback
inet addr:127.0.0.1  Bcast:127.255.255.255  Mask:255.0.0.0
UP BROADCAST LOOPBACK RUNNING  MTU:3584  Metric:1
RX packets:146 errors:0 dropped:0 overruns:0 frame:0
TX packets:146 errors:0 dropped:0 overruns:0 carrier:0
collisions:0

eth0 Link encap:Ethernet  HWaddr 00:80:C8:32:C8:1E
inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:864 errors:0 dropped:0 overruns:0 frame:0
TX packets:654 errors:0 dropped:0 overruns:0 carrier:0
collisions:0

Interrupt:10 Base address:0x6100
```

2 - Désactivez les 2 interfaces lo et eth0

ifconfig *lo* down

ifconfig *eth0* down

3 - Taper les commandes suivantes :

ping *localhost*

ping *192.168.1.1*

telnet *localhost*

Aucune commande ne fonctionne, car même si la configuration IP est correcte, les interfaces sont désactivées.

4 - Activez l'interface de loopback et tapez les commandes suivantes :

ifconfig *lo* up /* activation de l'interface de loopback */

ping *localhost* ou telnet localhost /* ça ne marche toujours pas */

route add *127.0.0.1* /* on ajoute une route sur l'interface de loopback */

ping *localhost* ou **telnet *localhost*** /* maintenant ça marche */

ping *192.168.1.1* /* ça ne marche pas car il manque encore une route*/

On peut déduire que :

- - pour chaque interface il faudra indiquer une route au protocole.
- - dans la configuration actuelle, aucun paquet ne va jusqu'à la carte, donc ne sort sur le réseau.

Voici le rôle de l'interface loopback. Elle permet de tester un programme utilisant le protocole IP sans envoyer de paquets sur le réseau. Si vous voulez écrire une application réseau, (telnet, ftp, ou autre), vous pouvez la tester de cette façon.

5 - Activez l'interface eth0 et tapez les commandes suivantes :

ifconfig *eth0* up /* activation de l'interface */

route add *192.168.1.1*

ifconfig /* l'information Tx/Rx de l'interface eth0 vaut 0 */

/* Aucun paquet n'est encore passé par la carte.*/

ping *127.0.0.1*

ifconfig /* on voit que l'information Tx/Rx de lo est modifiée */

/* pas celle de eth0, on en déduit que les paquets */

/* à destination de lo ne descendent pas jusqu'à l'interface physique */

ping *192.168.1.1* /* test d'une adresse locale */

ifconfig /* Ici on peut faire la même remarque. Les paquets ICMP */

/* sur une interface locale, ne sortent pas sur le réseau */

/* mais ceux de l'interface lo sont modifiés*/

ping ***192.168.1.2*** /* test d'une adresse distante */

ifconfig /* Ici les paquets sont bien sortis. Les registres TX/RX de eth0 */

/* sont modifiés, mais pas ceux de lo */

6 -Réalisez les manipulations suivantes, nous allons voir le comportement de la commande **ping** sur les interfaces.

Sur la machine tapez la commande

192.168.1.1 ifconfig /* relevez les valeurs des registres TX/RX */

192.168.1.2 ping 192.168.1.1

192.168.1.1 ifconfig /* relevez les nouvelles valeurs des registres TX/RX */

/* il y a bien eu échange Réception et envoi de paquets*/

192.168.1.2 ping 192.168.1.3

192.168.1.1 ifconfig /* On voit que le registre Rx est modifié mais */

/* le registre Tx n'est pas modifié. La machine a bien reçu*/

/* paquet mais n'a rien renvoyé */

192.168.1.2 ping 192.168.1.2

192.168.1.2 ifconfig /* aucun registre n'est modifié, donc les paquets */

/* ne circulent pas jusqu'à l'interface physique avec un .*/

/* ping sur l'interface locale */

7 - le MTU (*Message Transfert Unit*) détermine l'unité de transfert des paquets.

Vous allez, sur la machine 192.168.1.1 modifier le MTU par défaut à 1500, pour le mettre à 300, avec la commande :

ifconfig ***eth0 mtu 300***

Sur la machine d'adresse 192.168.1.2, vous allez ouvrir une session ftp et chronométrer le temps de transfert d'un fichier de 30 MO. Relevez le temps et le nombre de paquets transmis ou reçus (commande **ifconfig**, flags TX/RX).

Restaurez le paramètre par défaut sur la première machine.

Refaites le même transfert et comparez les chiffres. La différence n'est pas énorme sur le temps car le volume de données est peu important. Par contre la différence sur le nombre de paquets, elle, est importante.

5.3.2. La commande arp

Description de la commande

La commande **arp** permet de visualiser ou modifier la table du cache de l'interface. Cette table peut être statique et (ou) dynamique. Elle donne la correspondance entre une adresse IP et une adresse Ethernet.

À chaque nouvelle requête, le cache ARP de l'interface est mis à jour. Il y a un nouvel enregistrement. Cet enregistrement à une durée de vie (ttl ou *Time To Leave*).

Voici un exemple de cache ARP obtenu avec la commande **arp -va** :

```
? (192.168.1.2) at 00:40:33:2D:B5:DD [ether] on eth0
>Entries: 1      Skipped: 0      Found: 1
```

On voit l'adresse IP et l'adresse MAC correspondante. Il n'y a qu'une entrée dans la table. Voici les principales options de la commande **arp** :

arp -s (ajouter une entrée statique), exemple : **arp -s** ***192.168.1.2 00:40:33:2D:B5:DD***

arp -d (supprimer une entrée), exemple : **arp -d** ***192.168.1.2***

Voir la page **man** pour les autres options.

La table ARP et le fonctionnement d'un proxy ARP.

Cela est réalisé par la configuration de tables ARP statiques.

Le proxy est une machine qui est en interface entre un réseau et l'accès à Internet. Il fait office de passerelle et de cache à la fois.

- Passerelle, parce que tous les accès à Internet passent par le Proxy,
- Cache, parce que le Proxy conserve en mémoire cache (sur disque), une copie des pages consultées par les utilisateurs du réseau. Cela évite de télécharger à nouveau la même page sur le site d'origine, si un utilisateur revient fréquemment dessus.

Si un hôte du réseau demande l'adresse d'un noeud distant situé sur un autre réseau, et que cet hôte passe par un proxy, le proxy va renvoyer à l'hôte sa propre adresse Ethernet. Une fois cette opération réalisée, tous les paquets envoyés par l'hôte seront à destination de l'adresse Ethernet du proxy. Le proxy aura en charge de transmettre ces paquets à l'adresse effective du noeud distant.

Pour les réponses, un processus identique est mis en place. Le site consulté, ne retourne les réponses qu'au serveur proxy. Le serveur proxy se charge de ventiler les pages au bon destinataire du réseau local.

Voir, pour le fonctionnement des serveurs cache et la configuration des navigateurs avec ce type de serveur, le document sur le W3 et les scripts CGI.

Mode d'utilisation :

Attention à certaines interprétations de ce paragraphe. Il dépend de votre configuration. Soit vous êtes en réseau local avec une plage d'adresse déclarée, soit vous utilisez une carte d'accès distant.

Première partie :

1. Affichez le contenu de la table ARP avec la commande **`arp -a`**,
2. Supprimez chaque ligne avec la commande **`arp -d @ip`**, où *`@ip`* est l'adresse IP de chaque hôte apparaissant dans la table,
3. La commande **`arp -a`** ne devrait plus afficher de ligne,
4. Faites un **ping**, sur une station du réseau local,
5. **`arp -a`**, affiche la nouvelle entrée de la table,
6. Ouvrez une session sur Internet, puis ouvrez une session ftp anonyme sur un serveur distant en utilisant le nom, par exemple `ftp.cdrom.com`. Utilisez une adresse que vous n'avez jamais utilisée, supprimez également tout gestionnaire de cache.

7. Affichez le nouveau contenu de la table avec **`arp -a`**. Le cache ARP ne contient pas l'adresse Ethernet du site distant, mais celle de la passerelle par défaut. Cela signifie que le client n'a pas à connaître les adresses Ethernet des hôtes étrangers au réseau local, mais uniquement l'adresse de la passerelle. Les paquets sont ensuite pris en charge par les routeurs.
8. Refaites une tentative sur le site choisi précédemment. Le temps d'ouverture de session est normalement plus court. Cela est justifié, car les serveurs de noms ont maintenant dans leur cache la correspondance entre le nom et l'adresse IP.

Deuxième partie :

La commande **arp** permet de diagnostiquer un dysfonctionnement quand une machine prend l'adresse IP d'une autre machine.

1. Sur la machine 192.168.1.1, faites un **ping** sur 2 hôtes du réseau 192.168.1.2 et 192.168.1.3,
2. À l'aide de la commande **arp**, relevez les adresses MAC de ces noeuds,
3. Modifiez l'adresse IP de la machine 192.168.1.2 en 192.168.1.3
4. relancez les 2 machines en vous arrangeant pour que la machine dont vous avez modifié l'adresse ait redémarré la première,
5. Sur la machine d'adresse 192.168.1.1, remettez à jour les tables ARP.
6. Quel est le contenu, après cela de la table ARP ?

Conclusion : vous allez avoir un conflit d'adresses. Vous allez pouvoir le détecter avec la commande **arp**. Autre problème, si vous faites un **telnet** sur 192.168.1.3, il y a de fortes chances pour que ce soit la machine qui était d'adresse 192.168.1.2, qui vous ouvre la session. Nous sommes (par une action volontaire bien sûr) arrivés à mettre la pagaille sur un réseau de 3 postes. Cette pagaille pourrait tourner vite au chaos sur un grand réseau, d'où la nécessité pour un administrateur de faire preuve d'une grande rigueur.

Où en suis-je ?

Exercice 1 :

Vous êtes sur un réseau d'adresse 192.168.1.0 avec une interface d'adresse MAC 00:40:33:2D:B5:DD,

Vous n'avez aucun fichier host sur votre machine,

Il n'y a pas de DNS

La passerelle par défaut est 192.168.1.9

Vous faites un **`ping`** ***`195.6.2.3`*** qui a une interface d'adresse MAC 00:45:2D:33:C2 est localisée sur Internet

Le réseau fonctionne parfaitement et tout est parfaitement configuré

Cochez la bonne réponse:

A - On a dans la table arp ? (192.168.1.2) at 00:40:33:2D:B5:DD [ether] on eth0

B - On a dans la table arp ? (192.168.1.2) at 00:45:2D:33:C2 [ether] on eth0

C - On a dans la table arp ? (195.6.2.3) at 00:40:33:2D:B5:DD [ether] on eth0

D - On a dans la table arp ? (195.6.2.3) at 00: 00:45:2D:33:C2 [ether] on eth0

E - Il faut un fichier `host`, ou DNS pour réaliser l'opération **ping** demandée

F - Il n'est pas possible dans la configuration actuelle d'atteindre l'hôte 195.6.2.3

Réponse F, car la plage d'adresse 192.168.1.1 à 192.168.1.254 n'est pas routée sur l'Internet, sinon vous auriez l'adresse de la passerelle par défaut dans le cache ARP.

Exercice 2 :

Vous êtes sur un réseau d'adresse 192.5.1.0 avec une interface d'adresse MAC 00:40:33:2D:B5:DD,

Vous n'avez aucun fichier host sur votre machine,

Il n'y a pas de DNS,

La passerelle par défaut est 192.5.1.9

Vous faites un **`ping`** ***`www.existe.org`*** dont l'adresse IP est 195.6.2.3, et qui a une interface d'adresse MAC 00:45:2D:33:C2

Le réseau fonctionne parfaitement et tout est parfaitement configuré

Cochez la bonne réponse:

A - On a dans la table arp ? (192.5.1.0) at 00:40:33:2D:B5:DD [ether] on eth0

B - On a dans la table arp ? (192.5.1.0) at 00:45:2D:33:C2 [ether] on eth0

C - On a dans la table arp ? (195.6.2.3) at 00:40:33:2D:B5:DD [ether] on eth0

D - On a dans la table arp ? (195.6.2.3) at 00: 00:45:2D:33:C2 [ether] on eth0

E - Il faut un fichier `host`, ou DNS pour réaliser l'opération **ping** demandée

F - Il n'est pas possible dans la configuration actuelle d'atteindre l'hôte 195.6.2.3

Réponse E, car la résolution de noms ne peut être effectuée

Exercice 3 :

Vous êtes sur un réseau d'adresse 192.5.1.0, sur une machine d'adresse 192.5.1.1, et une interface d'adresse MAC 00:40:33:2D:B5:DD,

Vous n'avez aucun fichier `host` sur votre machine,

Il n'y a pas de DNS

La passerelle par défaut est 192.5.1.9, d'adresse MAC 09:44:3C:DA:3C:04

Vous faites un **`ping`** ***`195.6.2.3`***, et qui a une interface d'adresse MAC 00:45:2D:33:C2

Le réseau fonctionne parfaitement et tout est parfaitement configuré

Cochez la bonne réponse:

A - On a dans la table arp ? (192.5.1.0) at 00:40:33:2D:B5:DD [ether] on eth0

B - On a dans la table arp ? (192.5.1.0) at 00:45:2D:33:C2 [ether] on eth0

C - On a dans la table arp ? (195.6.2.3) at 00:40:33:2D:B5:DD [ether] on eth0

D - On a dans la table arp ? (192.5.1.9) at 09:44:3C:DA:3C:04 [ether] on eth0

E - Il faut un fichier `host`, ou DNS pour réaliser l'opération **ping** demandée

F - Il n'est pas possible dans la configuration actuelle d'atteindre l'hôte 195.6.2.3

Réponse D, l'hôte a bien été trouvé, la table ARP a été mise à jour avec l'adresse IP de la passerelle par défaut et son adresse Ethernet.

5.3.3. La commande route

La commande **route** a déjà été entrevue un peu plus haut, avec la commande **ifconfig**. Le routage définit le chemin emprunté par les paquets entre son point de départ et son point d'arrivée. Cette commande permet également la configuration de pc, de switchs de routeurs.

Il existe 2 types de routages :

- le routage statique

- le routage dynamique.

Le routage statique consiste à imposer aux paquets la route à suivre.

Le routage dynamique met en oeuvre des algorithmes, qui permettent aux routeurs d'ajuster les tables de routage en fonction de leur connaissance de la topologie du réseau. Cette actualisation est réalisée par la réception des messages reçus des noeuds (routeurs) adjacents.

Le routage dynamique permet d'avoir des routes toujours optimisées, en fonction de l'état du réseau (nouveaux routeurs, engorgements, pannes)

On combine en général le routage statique sur les réseaux locaux au routage dynamique sur les réseaux importants ou étendus.

Un administrateur qui dispose par exemple de 2 routeurs sur un réseau, peut équilibrer la charge en répartissant un partie du flux sur un port avec une route, et une autre partie sur le deuxième routeur.

Exemple de table de routage :

```
Kernel IP routing table
Destination  Gateway  Genmask   Flags  Metric  Ref Use  Iface
```

```
192.168.1.0  *   255.255.255. 0U     0       0   2    eth0
127.0.0.0  *   255.0.0.0    U      0       0   2    lo
default   192.168.1.9  0.0.0.0    UG     0       0   10   eth0
```

Commentaire généraux :

Destination : adresse de destination de la route

Gateway : adresse IP de la passerelle pour atteindre la route, * sinon

Genmask : masque à utiliser.

Flags : indicateur d'état (U - Up, H - Host - G - Gateway, D - Dynamic, M - Modified)

Metric : coût métrique de la route (0 par défaut)

Ref : nombre de routes qui dépendent de celle-ci

Use : nombre d'utilisation dans la table de routage

Iface : interface eth0, eth1, lo

Commentaire sur la 3ème ligne :

Cette ligne signifie que pour atteindre tous les réseaux inconnus, la route par défaut porte l'adresse 192.168.1.9. C'est la passerelle par défaut, d'où le sigle UG, G pour gateway.

Ajout ou suppression d'une route :

route add [net | host] addr [gw passerelle] [métric coût] [netmask masque] [dev interface]

- *net ou host* indique l'adresse de réseau ou de l'hôte pour lequel on établit une route,

- adresse de destination,

- adresse de la passerelle,

- valeur métrique de la route,

- masque de la route à ajouter,

- interface réseau à qui on associe la route.

Exemples :

route add *127.0.0.1 lo* /* ajoute une route pour l'adresse 127.0.0.1 sur l'interface lo */

route add -net *192.168.2.0 eth0* /* ajoute une route pour le réseau 192.168.2.0 sur l'interface eth0 */

route add *saturne.foo.org* /* ajoute une route pour la machine machin sur l'interface eth0 */

route add default gw *ariane* /* ajoute ariane comme route par défaut pour la machine locale */

/* ariane est le nom d'hôte d'un routeur ou d'une passerelle */

/* gw est un mot réservé */

route add *duschmoll* netmask *255.255.255.192*

/* Encore un qui a créé des sous réseaux., Il s'agit ici d'une classe c */

/* avec 2 sous réseaux, il faut indiquer le masque. */

Suppression d'une route :

route del -net *192.168.1.0*

route del -net *toutbet-net*

Avertissement

Attention, si on utilise des noms de réseau ou des noms d'hôtes, il faut qu'à ces noms soient associés les adresses de réseau ou des adresses IP dans le fichier `/etc/networks` pour les réseaux, et `/etc/hosts` ou DNS pour les noms d'hôtes.

Vous pouvez également voir l'atelier sur la mise en place d'un routeur logiciel.

Petite étude de cas :

Première partie - réalisation d'une maquette

On dispose de 2 réseaux (A et B) reliés par une passerelle. Le réseau A est également relié à Internet par un routeur. Le réseau A dispose d'un serveur de noms. Chaque réseau a deux machines.

```
Réseau  Nom du réseau Machine Nom des machines
A metaux-net 192.3.2.2 platine
192.3.2.3 uranium
192.3.2.4   mercure (serveur de noms)
B roches-net 130.2.0.2 quartz
130.2.0.3 silex
```

La passerelle entre le réseau A et B à 2 interfaces :

- eth0 192.3.2.1

- eth1 130.2.0.1

Le réseau A, a une passerelle par défaut pour Internet 130.2.0.9, qui est l'interface d'un autre routeur.

On veut :

- que les stations de chaque réseau puissent accéder à Internet,

- que les stations de chaque réseau puissent communiquer entre-elles,

- que les stations du réseau B, utilisent le serveur de noms le moins possible.

On demande :

1 - d'expliquer comment seront configurés les postes du réseau B,

2 - de donner la configuration des fichiers suivants pour chaque machine (`hosts`, `resolv.conf`, fichier de configuration de carte).

3 - de donner la liste des routes à mettre :

- sur les postes du réseau B,

- sur les postes du réseau A,

- sur la passerelle qui relie les 2 réseaux,

- sur le routeur du réseau A.

5.3.4. La commande netstat

La commande **netstat**, permet de tester la configuration du réseau, visualiser l'état des connexions, établir des statistiques, notamment pour surveiller les serveurs.

Liste des paramètres utilisables avec **netstat** :

Sans argument, donne l'état des connexions,

`-a` afficher toutes les informations sur l'état des connexions,

`-i` affichage des statistiques,

`-c` rafraîchissement périodique de l'état du réseau,

`-n` affichage des informations en mode numérique sur l'état des connexions,

`-r` affichage des tables de routage,

`-t` informations sur les sockets TCP

`-u` informations sur les sockets UDP.

État des connexions réseau avec **netstat**, dont voici un exemple :

```
 Proto  Recv-Q  Send-Q  Local Address   Foreign Address  State
 Tcp  0   126   uranus.planete.n:telnet 192.168.1.2:1037 ESTABLISHED
 Udp    0       0       uranus.plan:netbios-dgm  *:*
 Udp    0       0       uranus.plane:netbios-ns  *:*

Active UNIX domain sockets (w/o servers)
 Proto  RefCnt  Flags        Type        State         I-Node Path
 unix   2       [ ]          STREAM      1990     /dev/log
 unix   2       [ ]          STREAM CONNECTED 1989
 unix   1       [ ]          DGRAM        1955
```

Explications sur la première partie qui affiche l'état des connexions :

Proto : Protocole utilisé

Recv-q : nbre de bits en réception pour ce socket

Send-q : nbre de bits envoyés

LocalAdress : nom d'hôte local et port

ForeignAdress : nom d'hôte distant et port

State : état de la connexion

Le champ state peut prendre les valeurs suivantes:

Established : connexion établie

Syn snet : le socket essaie de se connecter

Syn recv : le socket a été fermé

Fin wait2 : la connexion a été fermée

Closed : le socket n'est pas utilisé

Close wait : l'hôte distant a fermé la connexion; Fermeture locale en attente.

Last ack : attente de confirmation de la fermeture de la connexion distante

Listen : écoute en attendant une connexion externe.

Unknown : état du socket inconnu

Explications sur la deuxième partie qui affiche l'état des sockets (IPC - *Inter Processus Communication*) actifs :

Proto : Protocole, en général UNIX,

Refcnt : Nombre de processus associés au socket

Type : Mode d'accès datagramme (DGRAM), flux orienté connexion (STREAM), brut (RAW), livraison fiable des messages (RDM)

State : Free, Listening, Unconnected, connecting, disconnecting, unknown

Path : Chemin utilisé par les processus pour utiliser le socket.

Affichage et état des tables de routage avec netstat : ***netstat -nr*** *ou* ***netstat -r***

```
Kernel IP routing table
Destination   Gateway   Genmask         Flags   MSS    Window  irtt  Iface
192.168.1.0   *         255.255.255.0   U       1500   0       0     eth0
127.0.0.0     *         255.0.0.0       U       3584   0       0     lo
```

Explications sur la commande ***netstat -r***

Destination : adresse vers laquelle sont destinés les paquets

Gateway : passerelle utilisée, * sinon

Flags : G la route utilise une passerelle, U l'interface est active, H on ne peut joindre qu'un simple hôte par cette route)

Iface : interface sur laquelle est positionnée la route.

Affichage de statistiques avec ***netstat -i***

```
Iface MTU  Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flags
Lo    3584 0   89    0      0      0      89    0      0      0      BLRU
eth0  1500 0   215   0      0      0      210   0      0      0      BRU
```

Explications sur la commande ***netstat -i***

RX-OK et TX-OK rendent compte du nombre de paquets reçus ou émis,

RX-ERR ou TX-ERR nombre de paquets reçus ou transmis avec erreur,

RX-DRP ou TX-DRP nombre de paquets éliminés,

RX-OVR ou TX-OVR recouvrement, donc perdus à cause d'un débit trop important.

Les Flags (B adresse de diffusion, L interface de loopback, M tous les paquets sont reçus, O arp est hors service, P connexion point à point, R interface en fonctionnement, U interface en service)

Exercices :

On donne les résultats de 3 commandes netstat ci-dessous, extraites de la même machine :

$ netstat -nr

Kernel IP routing table

Destination Gateway Genmask Flags MSS Window irtt Iface

198.5.203.0 0.0.0.0 255.255.255.0 U 1500 0 0 eth0

127.0.0.0 0.0.0.0 255.0.0.0 U 3584 0 0 lo

0.0.0.0 198.5.203.3 0.0.0.0 UG 1500 0 0 eth0

$ netstat

Active Internet connections (w/o servers)

Proto Recv-Q Send-Q Local Address Foreign Address State

Tcp 0 127 uranus.toutbet:telnet 194.206.6.143:1027 ESTABLISHED

$ netstat -i

Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flags

Lo 3584 0 764 0 0 764 89 0 0 0 BLRU

eth0 1500 0 410856 0 0 33286 210 0 0 0 BRU

On demande :

1. Quels sont les noms et adresse de la machine consultée ?
2. Quel type de session est-elle en train de supporter ?
3. À quoi correspond l'adresse 198.5.203.3 ?
4. Pourquoi une interface porte-t-elle les Flags BLRU et l'autre BRU ?
5. Quelle est la taille des paquets utilisée par la passerelle par défaut ?

5.3.5. La commande traceroute

La commande **traceroute** permet d'afficher le chemin parcouru par un paquet pour arriver à destination. Cette commande est importante, car elle permet d'équilibrer la charge d'un réseau, en optimisant les routes.

Voici le résultat de la commande **traceroute *`www.nat.fr`***, tapée depuis ma machine.

```
traceroute to sancy.nat.fr (212.208.83.2), 30 hops max, 40 byte packets
 1  195.5.203.9 (195.5.203.9)  1.363 ms  1.259 ms  1.270 ms
 2  194.79.184.33 (194.79.184.33)  25.078 ms  25.120 ms  25.085 ms
 3  194.79.128.21 (194.79.128.21)  88.915 ms  101.191 ms  88.571 ms
 4  cisco-eth0.frontal-gw.internext.fr (194.79.190.126)  124.796 ms[]
 5  sfinx-paris.remote-gw.internext.fr (194.79.190.250)  100.180 ms[]
 6  Internetway.gix-paris.ft.NET (194.68.129.236)  98.471 ms       []
 7  513.HSSI0-513.BACK1.PAR1.inetway.NET (194.98.1.214)  137.196 ms[]
 8  602.HSSI6-602.BACK1.NAN1.inetway.NET (194.98.1.194)  101.129 ms[]
 9  FE6-0.BORD1.NAN1.inetway.NET (194.53.76.228)  105.110 ms       []
10  194.98.81.21 (194.98.81.21)  175.933 ms  152.779 ms  128.618 ms[]
11  sancy.nat.fr (212.208.83.2)  211.387 ms  162.559 ms  151.385 ms[]
```

Explications :

Ligne 0 : le programme signale qu'il n'affichera que les 30 premiers sauts, et que la machine www du domaine nat.fr, porte le nom effectif de `sancy`, dans la base d'annuaire du DNS du domaine nat.fr. Cette machine porte l'adresse IP 212.208.83.2. Pour chaque tronçon, on a également le temps maximum, moyen et minimum de parcours du tronçon.

Ensuite, on a pour chaque ligne, l'adresse du routeur que le paquet a traversé pour passer sur le réseau suivant.

Ligne 4 et 5, le paquet a traversé 2 routeurs sur le même réseau 194.79.190.

Ligne 4, 5, 6, 7, 8, 9, 11, on voit que les routeurs ont un enregistrement de type A dans les serveurs de noms, puisqu'on voit les noms affichés.

Conclusion : depuis ma machine, chaque requête HTTP passe par 11 routeurs pour accéder au serveur www.nat.fr.

L'accès sur cet exemple est réalisé sur Internet. Un administrateur, responsable d'un réseau d'entreprise sur lequel il y a de nombreux routeurs, peut, avec cet outil, diagnostiquer les routes et temps de routage. Il peut ainsi optimiser les trajets et temps de réponse.

5.3.6. La commande dig

La commande **dig** remplace ce qui était la commande **nslookup**. Cette commande sert à diagnostiquer des dysfonctionnements dans la résolution de nom. (Service DNS).

Utilisation simple de **dig** :

```
$ dig any freenix.org
; <<>> DiG 9.2.2 <<>> any freenix.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21163
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 3, ADDITIONAL: 3

;; QUESTION SECTION:
;freenix.org.                   IN      ANY

;; ANSWER SECTION:
freenix.org.            92341   IN      SOA     ns2.freenix.org.\
                                         hostmaster.freenix.org.\
                                                     2003042501\
                                                          21600\
                                                           7200\
                                                        3600000\
                                                         259200\

freenix.org.            117930  IN      NS      ns2.freenix.fr.
freenix.org.            117930  IN      NS      ns.frmug.org.
freenix.org.            117930  IN      NS      ns6.gandi.net.

;; AUTHORITY SECTION:
freenix.org.            117930  IN      NS      ns2.freenix.fr.
freenix.org.            117930  IN      NS      ns.frmug.org.
freenix.org.            117930  IN      NS      ns6.gandi.net.

;; ADDITIONAL SECTION:
ns2.freenix.fr.         16778   IN      A       194.117.194.82
ns.frmug.org.           40974   IN      A       193.56.58.113
ns6.gandi.net.          259119  IN      A       80.67.173.196

;; Query time: 197 msec
;; SERVER: 213.36.80.1#53(213.36.80.1)
```

```
;; WHEN: Tue May 27 15:16:23 2003
;; MSG SIZE  rcvd: 248
```

retourne les informations sur le domaine concerné.

Il est ensuite possible d'intérroger sur tout type d'enregistrement : SOA, MX, A, CNAME, PTR...

5.3.7. La commande host

La commande **host** interroge les serveurs de coms. Elle peut par exemple être utilisée pour détecter des dysfonctionnement sur un réseau (serveurs hors services). Attention, n'utilisez pas cette commande sur des réseaux dont vous n'avez pas l'administration.

Chapitre 6. Les éditeurs joe et Emacs

Commandes de base pour pouvoir modifier les fichiers de configuration.

6.1. Présentation

Ce document donne les principales commandes qui permettent de commencer à utiliser un éditeur sous Linux. Emacs et Joe sont des éditeurs très utilisés sous Linux. Ils prennent peu à peu le pas sur VI (prononcer vi aïe). Sous Xwindow vous pouvez également utiliser Xemacs. Ces éditeurs sont normalement installés avec l'installation de Linux. Si cela n'est pas le cas, il vous faudra les installer ultérieurement. Les éditeurs sont les principaux outils utilisés pour la création de scripts ou de programmes sources. Leur principale différence avec un traitement de texte est qu'ils ne mettent aucun caractère de contrôle dans le document. Vous n'avez pas la possibilité de mettre en gras, italique, souligné. Pour installer par exemple l'éditeur joe, copiez le programme `joe-2.8-9.i386.rpm` de votre CD ROM sur le disque dur dans `/temp`. Installez-le avec la commande **`rpm -i joe-2.8-9.i386.rpm`**. Le programme joe est maintenant installé dans le répertoire `/usr/bin`. Vous pouvez l'utiliser en tapant **`joe`**.

6.2. L'éditeur Joe

```
Pour opbtenir de l'aide CTRL h

Les commandes de base
Rechercher CTRL k f
Rechercher suivant CTRL k l
Copier un block
Début de block CTRL k b
Fin de block CTRL k k
Copier le block CTRL k c
Déplacer le block CTRL k m
Supprimer le block CTRL k y
Ecran précédent CTRL u
Ecran suivant CTRL v
Début de document CTRL k u
Fin de document CTRL k v
Début de ligne CTRL a
Fin de ligne  CTRL e
Sauvegarder et quitter CTRL k x
Sauvegarder CTRL k d
Lire un fichier CTRL k e
Insérer un fichier CTRL k r
Accéder au shell  CTRL k z (taper fg pour revenir)
Quitter  CTRL x c
```

6.3. L’éditeur Emacs

```
Notation des touches :
CTRL : signifie Ctrl
META : signifie Alt
ESC : signifie ECHAP
SHT : signifie SHIFT
RET : return
SPB : signifie barre d’espace

Les commandes de base :
Lancement de Emacs :
emacs : lancement avec un fichier vide
emacs NomFichier : édite le Fichier de nom NomFichier

Action   Touches
Accéder à l’aide  CTRL h
Répertoire : liste    CTRL x  CTRL d
Annuler Cmd en cours  CTRL g
Annuler cmd précédente    CTRL x  u
Annuler modifications     ESC  ~
Curseur End    CTRL e
Curseur Home  CTRL a
Reculer d’un caractère   CTRL b
Avancer d’un caractère    CTRL f
Défilement PgDn   CTRL v
Défilement PgUp   ESC  v
Effacer caractère droite CTRL d
Effacer fin de ligne  CTRL k
Fichier : charger     CTRL x  CTRL f
Fichier : insérer     CTRL x  CTRL i
Fichier : sauver  CTRL x  CTRL s
Fichier : (re)nommer  CTRL x CTRL w nom
Positionnement haut   ESC  <
Positionnement bas    ESC  >
Rechercher    CTRL s
Remplacer     ESC  %
Bloc: marque debut     CTRL SPB
Coller region (paste)     CTRL y
Copier region (copy)  ESC  w
Couper region (cut)   CTRL w
Aller à la ligne ..   ESC  x
Quitter   CTRL x  CTRL c
Annuler une commande CTRL g

Gestion des fenêtres et des Buffers
Liste des buffers CTRL x  CTRL b
Changer de fenêtre CTRL x o
Maximiser la fenêtre courante CTRL x 1
```

6.4. L’incontournable vi

```
Commande Description

0 Aller en début de ligne
$ Aller en fin de ligne
k Ligne précédente
j Ligne suivante
h Caractère précédent
l Caractère suivant
b Mot précédent
w Mot suivant
Ctrl-B Page précédente
Ctrl-F Page suivante
nG Aller à la ligne n. Exemple: 1G va sur la première ligne
G Aller à la dernière ligne
x Supprime le caractère sous le curseur
dd Supprime la ligne courante et la copie dans le presse-papiers
nd Idem avec n lignes
J Fusionne la ligne courante et la suivante
yy Copie la ligne courante dans le presse-papiers
ny Idem avec n lignes
P Colle le presse-papiers avant la position courante
p Colle le presse-papiers après la position courante
v Commence une sélection en mode caractères
V Commence une sélection en mode lignes
Ctrl-V Commence une sélection en mode "rectangulaire"
d Supprime la sélection et la copie dans le presse-papiers
y Copie la sélection dans le presse-papiers
c Supprime la sélection et passe en mode insertion
i Passe en mode insertion avant la position courante
a Passe en mode insertion après la position courante
o Passe en mode insertion sur une nouvelle ligne sous la ligne courante
Esc Quitte le mode insertion
u Annule la dernière commande
r Remplace le caractère sous le curseur par le prochain caractère tapé
~ Convertit le caractère sous le curseur en majuscule si c’est une minuscule et vice-vers
/texte Recherche en avant du texte indiqué
?texte Recherche en arrière du texte indiqué
n Recherche l’occurrence suivante
N Recherche l’occurrence précédente
:%s/find/replace Recherche avec remplacement dans tout le fichier
:w Sauvegarde le fichier courant
:wfichier Ecrit le document dans le fichier indiqué
:rfichier Inclut le fichier indiqué à partir de la position courante
:q! Quitter en annulant les modifications
ZZ (ou :wq) Quitter en enregistrant les modifications
```